OMNIINVENT: GENERALIST GENERATIVE CHEMICAL LANGUAGE MODELS THROUGH BLANK INFILLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Drug design requires navigating a vast and sparsely populated chemical space while balancing structural novelty with multiple property constraints. Most generative approaches to the problem remain specialized to individual tasks, however. Here, we introduce OmniInvent, a chemical language model that operates on unmodified SMILES and supports both de novo and substructure-constrained molecular generation within a single framework. OmniInvent formulates molecular design as an autoregressive blank infilling problem over partially masked SMILES and is pretrained using an on-the-fly graph-based masking strategy that randomly selects connected molecular substructures and masks the corresponding SMILES segments without the need for task-specific fragmentation strategies. In direct sampling, OmniInvent generates highly valid, unique, and diverse molecules in de novo design, fragment linking, scaffold decoration, and motif extension. In goal-directed design, it matches or exceeds state-of-the-art performance on the Practical Molecular Optimization benchmark for de novo design and achieves competitive performance in substructure-constrained design despite being trained as a general-purpose model. These results show that blank infilling on SMILES provides an effective foundation for building generalist molecular generative models across diverse drug design tasks.

1 INTRODUCTION

Drug discovery requires searching a vast and sparsely populated chemical space for molecules that satisfy multiple, often competing, objectives (Mandal et al., 2009). Generative machine learning models can support this process by proposing novel structures with desirable properties (Schneider et al., 2020; Muller et al., 2022). Chemical language models (CLMs), which represent molecules using line notations such as Simplified Molecular Input Line Entry System (SMILES) (Weininger, 1988), are a practical approach because they can learn chemical syntax and structure from large molecular datasets (Grisoni, 2023) and be guided towards generating molecules with optimized properties through techniques like reinforcement learning (RL) (Olivecrona et al., 2017), thus enabling goal-directed design.

Most SMILES-based molecular generators are specialized to a particular design setting. For example, autoregressive models can be used for de novo generation (Olivecrona et al., 2017), while scaffold decoration and fragment linking are commonly addressed using separately trained models with task-specific molecular decompositions and input representations (Fialková et al., 2022; Guo et al., 2023). This specialization introduces separate preprocessing and training pipelines, and limits the ability to transfer across design tasks. A generalist generator should instead use a common representation of molecular constraints and generate the missing content regardless of whether the task is de novo design or substructure-constrained design, e.g. fragment linking, scaffold decoration, or motif extension.

Recent developments in CLMs have shown that a single model can support multiple design tasks by operating on fragment-based representations such as SAFE (Noutahi et al., 2024; Lee et al., 2025; Kaech et al., 2026). These representations require explicit fragmentation and reassembly conventions, however. Here, we investigate an alternative formulation that operates directly on unmodified SMILES (Weininger, 1988). Our contributions are threefold: we formulate de novo and substructure-constrained molecular design as autoregressive blank infilling over standard SMILES;

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.

introduce OmniInvent, combining prefix attention, two-dimensional positional embeddings, and graph-based on-the-fly masking; and demonstrate competitive unguided and RL-guided generation across de novo design as well as substructure-constrained design.

2 Related work

Chemical language models for drug design. Beyond graph-based (Jin et al., 2018; Mercado et al., 2021) and 3D (Hoogeboom et al., 2022; Irwin et al., 2025) generative models, drug design is commonly framed as sequence modeling over molecular line string representations such as SMILES (Weininger, 1988). Many of these so-called CLMs are autoregressive in nature and produce exact sequence log-likelihoods. This allows policy-gradient RL to be used to steer models towards generating molecules with desirable properties, i.e., goal-directed drug design. The REINVENT framework and related methods have demonstrated this capability in a de novo setting (Olivecrona et al., 2017; Blaschke et al., 2020). This was later extended to substructure-constrained design with LibINVENT (Fialková et al., 2022) and LinkINVENT (Guo et al., 2023), respectively designed for scaffold decoration and fragment linking, through task-specialized conditional generative models. These approaches are effective, but make different assumptions about the structure of input data, thus requiring separate training and data preprocessing pipelines for different design settings. SAMOA (Langevin et al., 2020) and PromptSMILES (Thomas et al., 2024) instead use specialized sampling techniques to conduct substructure-constrained molecular design using a backbone model trained for de novo generation.

Generalist molecular generation models. Recent work, including SAFE-GPT (Noutahi et al., 2024), GenMol (Lee et al., 2025), and InVirtuoGen (Kaech et al., 2026), supports both de novo design and substructure-constrained generation using fragment-based molecular string representations. These methods show that general-purpose molecular generators can be competitive across tasks. However, fragment-based representations require molecular fragmentation and thus rely on hand-crafted bond-breaking rules. OmniInvent instead operates on standard SMILES, avoiding fragmentation while retaining a common formulation for de novo and substructure-constrained design.

Blank infilling models. Various techniques have been developed to allow language models to conduct blank infilling, that is, to fill in blanks in a token sequence. Bavarian et al. (2022) accomplished this using a standard causal decoder transformer; their approach was tailored towards filling in single blanks, however, rendering multi-blank infilling difficult. Masked diffusion models (Sahoo et al., 2024), as well as similar adaptations of encoder transformers like FiLM (Shen et al., 2023), are capable of filling in an arbitrary number of blank spans in a given sequence, but typically require the number of tokens at each blank position to be pre-defined at inference time. Methods that accomplish multi-blank infilling that support autoregressive decoding have also been developed, including T5 (Raffel et al., 2020) and InCoder (Fried et al., 2023), which utilize unique sentinel tokens representing each blank. The General Language Model (GLM) of Du et al. (2022) supports autoregressive decoding and uses a similar approach; unlike T5 and InCoder, however, the same mask token is used for every blank. GLM instead utilizes prefix attention together with two-dimensional positional encodings to encode the positional information relevant to decoding. We adopt the GLM approach to build OmniInvent by formulating molecular design as a blank infilling problem over SMILES sequences, making RL-based molecular optimization possible.

3 Methods

3.1 Model input representation

Drug design may involve generating an entire molecule (de novo design) or a molecule subject to substructure constraints (substructure-constrained design). We cast these tasks as autoregressive blank infilling problems over tokenized SMILES sequences, using the tokenization scheme of Schwaller et al. (2018). Our approach to molecular design is illustrated in Figure 1. The full molecular graph is first mapped to the corresponding SMILES sequence. The SMILES sub-sequences corresponding to the molecular subgraphs that are to be replaced, such as the scaffold or a decorator, are substituted with mask tokens denoted `[MASK]`. OmniInvent subsequently generates the missing content conditioned on the partially masked SMILES sequence. In this example, a single mask to-

ken is sufficient for each subgraph; this is not always the case, however. When a subgraph maps to multiple disjoint spans in the SMILES sequence, each such span is replaced with `[MASK]`. In the de novo setting, the model is conditioned on a single mask token and autoregressively generates the full molecular representation.

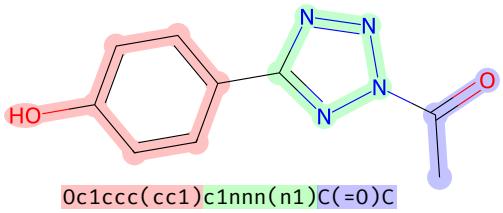


Figure 1: A SMILES sequence and its corresponding molecular graph. Three distinct subgraphs are highlighted: the scaffold (green) and two decorators (red/blue). Different drug design scenarios can be realized through autoregressive blank infilling by selectively masking out parts of the SMILES: fragment linking by masking the scaffold, motif extension by masking out a single decorator, and scaffold decoration by masking both decorators.

3.2 Model Architecture

OmniInvent is based on the LLaMA architecture (Touvron et al., 2023), a decoder transformer for large language models, as implemented in the `transformers` library (Wolf et al., 2020). Following the GLM approach of Du et al. (2022), we adapt the architecture by incorporating prefix attention (Dong et al., 2019; Raffel et al., 2020) and a two-dimensional Rotary Position Embedding (RoPE) variant (Su et al., 2024; Heo et al., 2025). These concepts are illustrated in Figure 2 and further expanded upon in Section A.

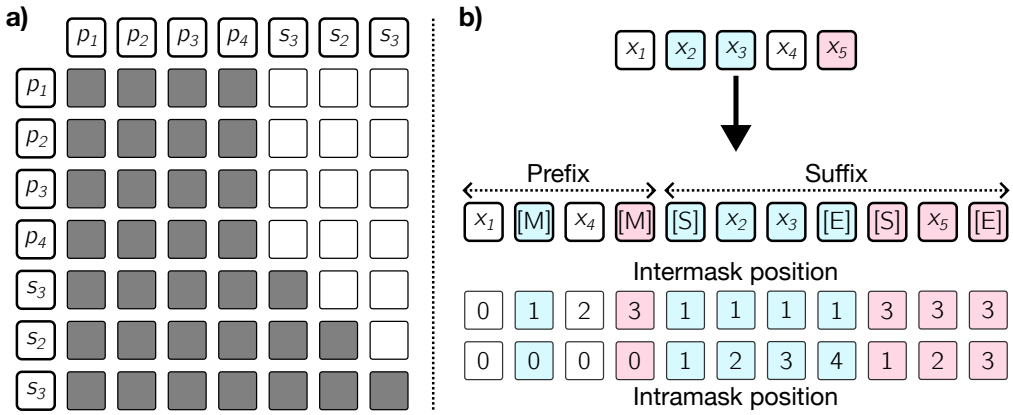


Figure 2: Illustrations of a) a prefix attention mask, wherein tokens in the prefix (p) attend to each other while tokens in the suffix (s) attend only to previous tokens, and b) two-dimensional position embeddings, allowing for the encoding of both the positions of mask tokens in the prefix (intermask) as well as sequential token positions in the prefix and suffix (intramask).

3.2.1 Model Pretraining

We train separate OmniInvent models on datasets derived from the ChEMBL 35 (Gaulton et al., 2012) as well as the PubChem Compound (Kim et al., 2016) databases. SMILES from the databases are first preprocessed: unwanted molecules are discarded and the representations of the remaining ones standardized. A random split is then conducted into training (90 %) and validation (10 %) sets; the latter is monitored during training to mitigate overfitting. The ChEMBL-derived training dataset contains 2 044 230 SMILES and the PubChem-derived training dataset 94 587 462 SMILES. Two model sizes are trained for each dataset, which we refer to as OmniInvent (28 million parameters)

and OmniInvent$_{\text{L}}$ (67 million parameters). Details on the preprocessing pipeline and model layouts may be found in Sections B and C, respectively.

Masked training examples for OmniInvent are generated on the fly using a graph-based masking procedure, thus avoiding the use of hand-crafted fragmentation rules. For a given input molecule, a fully masked SMILES is generated with probability p_{full}. Otherwise, between 1 and 4 random connected subgraphs of the molecular graph are chosen and mapped back to the corresponding SMILES sequence; each of these spans is wrapped with sequence start and end tokens and concatenated in a left-to-right order, forming the suffix. The spans in the original SMILES are replaced with `[MASK]`, which forms the prefix. The suffix is appended to the prefix to form the final training example as illustrated in Figure 2b). While the framework supports any-order infilling of mask tokens, they are treated in sequence order (left-to-right) here.

SMILES sequences are based on walks of the corresponding molecular graph (Weininger, 1988); they are thus non-unique, and a molecule generally has several equally valid SMILES representations. This has been used as a form of data augmentation for CLMs (Arús-Pous et al., 2019). To this end, we train separate models using one of two SMILES randomization strategies by utilizing the RDKit library: permutation randomization, based on randomizing the order of atomic indices in the library's internal molecular representation, and RDKit randomization, which makes use of the built-in randomization feature of the library. The two schemes were respectively called restricted and unrestricted by Arús-Pous et al. (2019), who also note that the former produces a subset of the randomizations produced by the latter method. A more detailed description of the input generation strategy may be found in Section D.1.

Models were trained with a schedule-free AdamW variant (Defazio et al., 2024) using standard cross-entropy loss applied to the suffix under teacher forcing. The training setups are described in detail in Section D.2. Two datasets, two model sizes, and two randomization strategies yields eight OmniInvent models; for brevity, the main text presents only the PubChem-trained variants with training-time RDKit randomization. Results for the other six may be found in the appendix. Additionally, we train the smaller OmniInvent variant on ZINC 250K (Irwin et al., 2012; Gómez-Bombarelli et al., 2018) using RDKit randomization for use in the goal-directed de novo design benchmark.

3.3 Evaluation

We evaluate OmniInvent on de novo design and substructure-constrained design in both unguided and goal-directed scenarios. The unguided case corresponds to direct sampling from the models, while RL was utilized in the goal-directed case to guide the model towards generating high-scoring molecules. In all our evaluation settings, we use multinomial sampling with a temperature of 1.

3.3.1 Direct sampling

In the unguided setting, OmniInvent generates SMILES sequences conditioned on a single `[MASK]` token. Following Noutahi et al. (2024), we report validity, uniqueness, and internal diversity for the generated samples. A generated SMILES is considered valid if it is successfully parsed and sanitized by RDKit. Uniqueness is defined as the fraction of valid non-duplicate SMILES generated during a single run after canonicalization. Internal diversity (hereafter, diversity) is defined as the average pairwise Tanimoto distance of the generated samples using Morgan fingerprints (2048-bit, radius 2).

In the substructure-constrained case, the model is conditioned on partially masked SMILES sequences representing molecular fragments. Model inputs are constructed by randomizing the SMILES representation of the fragments and inserting `[MASK]` tokens representing the blanks to be filled in, as described in Section E. Following Noutahi et al. (2024), we use fragments derived from ten known drugs for fragment linking, scaffold decoration, as well as motif extension. Motif extension can be viewed as scaffold decoration with a single attachment point. Following Lee et al. (2025), we generate 100 SMILES for each drug-task combination and compute validity, uniqueness, diversity, and distance to the source molecule. For each design scenario, we report these metrics averaged across the ten drugs. The last metric (hereafter, distance) is defined as the mean Tanimoto distance between the Morgan fingerprints (1024-bit, radius 2) of the unfragmented source molecule and those of the generated samples.

We note that the fingerprint settings above follow those used by Lee et al. (2025).

3.3.2 GOAL-DIRECTED DESIGN

To generate molecules with particular properties, i.e., conduct goal-directed design, we employ RL using a variant of the REINFORCE algorithm as proposed by Olivecrona et al. (2017) by interfacing with the REINVENT4 framework (Loeffler et al., 2024). For **de novo design**, we use the Practical Molecular Optimization (PMO) benchmark which consists of 23 different oracle functions (Gao et al., 2022) to be optimized under a budget of at most 10 000 oracle calls. Results are reported as the area under the curve (AUC) of the mean top-10 molecule scores as a function of the number of oracle calls. We additionally employ experience replay: the top-24 molecules generated so far are saved in a buffer and concatenated to the sample batch drawn from the model in each epoch.

Several recent fragment-based approaches have applied prescreening to the PMO benchmark (Lee et al., 2024; 2025; Kaech et al., 2026). Instead of generating molecules from scratch, these methods first score the ZINC 250K database using the oracles of interest. An initial set of fragments is then constructed based on their frequency in high-scoring molecules and subsequently combined to form new molecules. OmniInvent does not support this type of fragment-based design. Therefore, we instead perform additional experiments using the PubChem-trained OmniInvent variants with RDKit randomization; we prescreen ZINC 250K and populate the replay buffer with the 24 highest-scoring molecules before starting the RL loop.

For **substructure-constrained design**, we adapt tasks from Fialková et al. (2022) and Guo et al. (2023) for scaffold decoration and fragment linking, respectively. The scaffold decoration task involves optimizing dopamine receptor D_2 (DRD2) activity, predicted by a machine learning model from Olivecrona et al. (2017) as implemented in the Therapeutics Data Commons package (Huang et al., 2021), while avoiding undesired substructures; if the generated sample contains such a substructure, its score is set to zero. In fragment linking, the aim is to generate molecules with both physicochemical properties and linker length within specific ranges. The original task definition additionally required the generated linkers to be linear; we run evaluations both with and without this constraint.

Detailed descriptions of the experimental setups, including hyperparameters, are available in Section F.

4 EXPERIMENTS

We assessed OmniInvent's generative capabilities in direct sampling and goal-directed drug design, each under both de novo and substructure-constrained settings. For brevity, we present only results for the PubChem-trained model, which used training-time RDKit randomization. For the substructure-constrained experiments, input fragments are randomized with RDKit randomization for direct sampling and permutation randomization for the goal-directed setting. Results for the other pretraining and randomization settings may be found in Sections H.1 and H.2.

4.1 DIRECT SAMPLING

Direct sampling in a de novo setting was conducted by generating 10 000 molecules without input fragments; the results are shown in Table 1, with a set of randomly picked samples shown in Figure 8. We additionally included GenMol and SAFE-GPT from the work of Lee et al. (2025) as well as the public de novo generator integrated into REINVENT4 (REINVENT_4) (Olivecrona et al., 2017; Blaschke et al., 2020; Loeffler et al., 2024) as baselines. We omit InVirtuoGen from this comparison, as Kaech et al. (2026) presented de novo direct sampling results only graphically; its performance was similar to that of GenMol, however.

Performance differences between models were small; all of them were capable of generating valid, unique, and diverse samples, although samples from both OmniInvent variants exhibited marginally higher diversity than the rest. REINVENT_4 was the strongest baseline in terms of diversity, falling only slightly behind OmniInvent. Pretraining on ChEMBL yielded slightly lower validity and diversity as shown in Table 10, while both training-time RDKit and permutation randomization yielded similar results.

Table 1: Results from de novo direct sampling with OmniInvent, GenMol, and SAFE-GPT. For GenMol, Lee et al. (2025) reported results for several generation settings; here, we included the setting maximum uniqueness. OmniInvent, $\text{OmniInvent}_\text{L}$, and REINVENT_4 results were averaged over three runs, with errors indicating the standard deviation.

Method	Valid. (%)	Uniq. (%)	Diversity
OmniInvent	99.0 ± 0.0	100.0 ± 0.0	0.895 ± 0.001
$\text{OmniInvent}_\text{L}$	99.2 ± 0.0	100.0 ± 0.0	0.897 ± 0.001
REINVENT_4	98.3 ± 0.0	100.0 ± 0.0	0.888 ± 0.000
GenMol	99.8 ± 0.1	100.0 ± 0.0	0.845 ± 0.002
SAFE-GPT	94.0 ± 0.4	100.0 ± 0.0	0.879 ± 0.001

Table 2: Results from substructure constrained generation on fragments from ten drugs (Noutahi et al., 2024) for OmniInvent, GenMol, SAFE-GPT, and InVirtuoGen. OmniInvent input fragments were randomized using RDKit and results were averaged over five independent runs, with errors representing the standard deviation.

Method	Task	Valid. (%)	Uniq. (%)	Diversity	Distance
SAFE-GPT	Fragm. linking	76.6 ± 5.1	82.5 ± 1.9	0.545 ± 0.007	0.541 ± 0.006
	Scaffold dec.	97.7 ± 0.3	74.7 ± 2.5	0.575 ± 0.008	0.625 ± 0.009
	Motif ext.	96.1 ± 1.9	66.8 ± 1.2	0.562 ± 0.003	0.665 ± 0.006
GenMol	Fragm. linking	100.0 ± 0.0	83.7 ± 0.5	0.547 ± 0.002	0.563 ± 0.003
	Scaffold dec.	96.6 ± 0.9	82.7 ± 1.8	0.591 ± 0.001	0.651 ± 0.001
	Motif ext.	82.9 ± 0.1	77.5 ± 0.1	0.617 ± 0.002	0.682 ± 0.001
InVirtuoGen	Fragm. linking	60.4 ± 0.6	84.7 ± 1.6	0.52 ± 0.004	Not reported
	Scaffold dec.	90.7 ± 0.6	88.6 ± 1.1	0.56 ± 0.003	Not reported
	Motif ext.	69.0 ± 0.8	96.8 ± 0.3	0.62 ± 0.005	Not reported
OmniInvent	Fragm. linking	99.5 ± 0.2	83.8 ± 1.2	0.532 ± 0.003	0.524 ± 0.001
	Scaffold dec.	97.6 ± 0.6	93.8 ± 0.9	0.568 ± 0.005	0.474 ± 0.005
	Motif ext.	99.0 ± 0.3	97.4 ± 0.4	0.664 ± 0.003	0.656 ± 0.002
$\text{OmniInvent}_\text{L}$	Fragm. linking	99.4 ± 0.2	80.8 ± 1.1	0.530 ± 0.004	0.496 ± 0.005
	Scaffold dec.	98.0 ± 0.6	92.2 ± 1.3	0.553 ± 0.005	0.449 ± 0.006
	Motif ext.	99.3 ± 0.3	97.9 ± 0.9	0.656 ± 0.002	0.637 ± 0.001

For substructure-constrained generation, we evaluated fragment linking, scaffold decoration, and motif extension using the SAFE-GPT test set, which contains task-specific molecular fragments derived from ten drugs (Noutahi et al., 2024). Results are reported in Table 2, including SAFE-GPT (Noutahi et al., 2024), GenMol (Lee et al., 2025), and InVirtuoGen (Kaech et al., 2026). Random OmniInvent samples for one of the drugs may be found in Figure 9.

OmniInvent performed strongly across the board, producing highly valid and unique samples; in contrast, each baseline fell below 90 % validity for at least one task. Its sample diversity was comparable to GenMol's, exceeding it for the motif extension task and trailing it on scaffold decoration. Source molecule distances were lower than for the baselines that reported this metric; in other words, OmniInvent produced internally diverse sets of samples that were more similar to the source drug than those of the baselines. $\text{OmniInvent}_\text{L}$ performed comparably to the smaller variant, and pretraining on ChEMBL rather than PubChem gave similar results (Section H.2). The differences between inference-time RDKit and permutation randomization applied both during training and for model input construction were also small, with one exception: training-time permutation randomization with RDKit-randomized input fragments resulted in degraded performance. Representation mismatch is a likely explanation: the SMILES produced by permutation randomization are a subset of those produced by RDKit randomization, so fragments randomized with the latter may lie out of distribution for models trained with the former.

We note that OmniInvent's performance was sensitive to the input sequence. For the results in Table 2, we used the maximum possible number of mask tokens for each fragment set; reducing the num-

ber of masks decreased uniqueness, as shown in Table 11. Additionally, two scaffold decoration fragment sets contained more than four attachment points, thus falling outside OmniInvent's pretraining regime. Despite this mismatch, we observed no detrimental effect on overall performance.

4.2 GOAL-DIRECTED DRUG DESIGN

We evaluated OmniInvent's capability of performing de novo goal-directed design on the PMO benchmark suite, with summary results presented in Table 3. Additional results, including oracle-specific breakdowns and other OmniInvent variants, may be found in Section H.3. As baselines, we included results from Gao et al. (2022) ($\text{REINVENT}_{\text{PMO}}$), InVirtuoGen (Kaech et al., 2026), GenMol (Lee et al., 2025), f-RAG (Lee et al., 2024), and Genetic GFN (Kim et al., 2024). We also conducted additional experiments using the public de novo generator integrated into REINVENT4 (REINVENT_4), which differs from $\text{REINVENT}_{\text{PMO}}$ in two important ways: it uses a long-short term memory recurrent neural network rather than one based on gated recurrent units and was trained on a dataset derived from ChEMBL rather than ZINC 250K. The models in the original PMO benchmark comparison were pretrained on ZINC 250K (Gao et al., 2022); therefore, we pretrained the smaller OmniInvent variant on that dataset as well ($\text{OmniInvent}_{\text{ZINC}}$).

Table 3: Top-10 AUCs summed over the 23 PMO oracles for OmniInvent and baseline models. For OmniInvent and REINVENT_4, scores are reported as averages over five independent runs, with errors representing the standard deviation. The REINVENT_4 experiments were run using the same hyperparameter settings as $\text{REINVENT}_{\text{PMO}}$. Only InVirtuoGen reported standard deviations of the baselines.

Model (no prescreening)	Score	Model (prescreening)	Score
OmniInvent	16.734 ± 0.241	OmniInvent	18.641 ± 0.321
$\text{OmniInvent}_{\text{L}}$	16.861 ± 0.135	$\text{OmniInvent}_{\text{L}}$	18.879 ± 0.085
InVirtuoGen	16.676 ± 0.256	InVirtuoGen	18.993 ± 0.219
Genetic GFN	16.213	GenMol	18.362
$\text{OmniInvent}_{\text{ZINC}}$	15.257 ± 0.164	f-RAG	16.928
REINVENT_4	15.785 ± 0.118		
$\text{REINVENT}_{\text{PMO}}$	14.196		

Without prescreening, OmniInvent achieved one of the highest overall PMO scores reported in this setting to date, to our knowledge. Of the variants shown in Table 3, $\text{OmniInvent}_{\text{L}}$ scored highest, with the smaller variant performing similarly; the ChEMBL-pretrained variants and those pretrained with permutation randomization performed comparably as well. Direct sampling metrics showed no clear relationship with PMO benchmark score; pretraining on PubChem yielded slightly higher validity and diversity in de novo generation than ChEMBL pretraining, but the highest-scoring OmniInvent variants overall were pretrained on ChEMBL (Section H.3). This suggests that these metrics may be poor predictors of performance in goal-directed settings. Furthermore, $\text{OmniInvent}_{\text{L}}$ outscored the smaller variant in only one of four pretraining setups, potentially indicating limited gains from further model scaling; similar observations have been made for other classes of CLMs (Krüger et al., 2025; Medina et al., 2026). The strongest baseline was InVirtuoGen, scoring slightly below, but within one standard deviation of, OmniInvent, with Genetic GFN not far behind. InVirtuoGen was pretrained on the SAFE-GPT training dataset of over one billion molecules, however (Noutahi et al., 2024; Kaech et al., 2026). All OmniInvent variants were pretrained on substantially smaller datasets. As the lower score of $\text{OmniInvent}_{\text{ZINC}}$ relative to OmniInvent demonstrates, training dataset size can have a dramatic impact on the PMO benchmark. Genetic GFN yielded strong results despite being trained on ZINC 250K; it uses the $\text{REINVENT}_{\text{PMO}}$ architecture as its backbone but combines a genetic algorithm with a GFlowNet rather than RL to guide generation toward high-scoring molecules. Given that $\text{OmniInvent}_{\text{ZINC}}$ scored higher than $\text{REINVENT}_{\text{PMO}}$, using this genetic-guided GFlowNet approach with OmniInvent as a backbone is likely to further improve performance.

With prescreening, InVirtuoGen achieved the highest PMO score, with $\text{OmniInvent}_{\text{L}}$ scoring slightly lower, but within one standard deviation of the baseline. Both OmniInvent variants in Table 3 yielded scores higher than those of GenMol and f-RAG under these conditions. We emphasize that this setting is at odds with the PMO benchmark setup, however, as the prescreening oracle calls are not

accounted for (Gao et al., 2022). The approach nevertheless significantly increases PMO scores relative to the prescreening-free case, as illustrated by both InVirtuoGen and OmniInvent.

For goal-directed substructure-constrained design, we adapted scaffold decoration and fragment linking scenarios from Fialková et al. (2022) and Guo et al. (2023), respectively, as outlined in the Methods section. The results are shown in Figure 3 for the OmniInvent variants pretrained on PubChem with RDKit randomization. Permutation randomization was used for the model inputs. Results for ChEMBL pretraining and other randomization schemes may be found in Section H.4. To our knowledge, there is no publicly available code for running goal-directed design in substructure-constrained scenarios with SAFE-GPT, GenMol, or InVirtuoGen. Therefore, we instead used the publicly available LibINVENT (Fialková et al., 2022) and LinkINVENT (Guo et al., 2023) models integrated into REINVENT4 as baselines here. We note that these models are encoder-decoder transformers pretrained on PubChem rather than the models introduced with the respective publications.

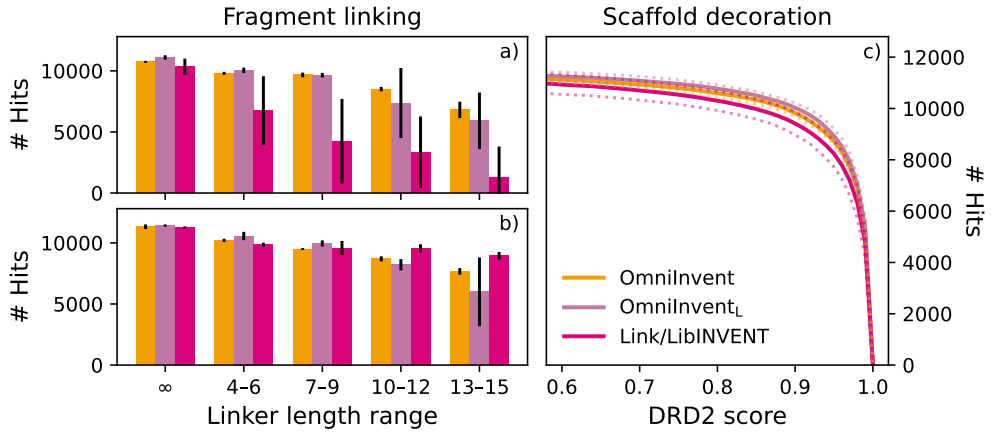


Figure 3: Average number of hits for a) fragment linking with linearity constraint for different target linker lengths, b) fragment linking without linker linearity constraint, and c) scaffold decoration. Hits are defined as SMILES that encode molecules fulfilling all scoring constraints. For a) and b), results are averaged over five independent runs, with error bars representing standard deviations; ∞ represents the case without linker length constraints. For c), results are presented as a function of DRD2 oracle score, i.e., the number of generated molecules with at least a certain score, and averaged over three independent runs, with the dotted lines representing the standard deviation between runs.

In fragment linking, LinkINVENT struggled to generate linear linkers (Figure 3a), while OmniInvent showed no such limitation even at longer linker lengths. This advantage disappeared when the linearity constraint was relaxed, however; both models then generated comparable numbers of linkers in every length range (Figure 3b). OmniInvent$_L$ performed similarly to the smaller model here, albeit with higher inter-run variance at longer linker lengths. For scaffold decoration, the two OmniInvent size variants performed comparably, producing slightly more hits than LibINVENT across the entire DRD2 oracle score range. We emphasize that OmniInvent was pretrained with a blank-infilling strategy on examples derived from masking random subgraphs, whereas LibINVENT and LinkINVENT were pretrained specifically for the two drug design tasks tested here.

5 Ablations

To investigate the impact of some key design choices with OmniInvent, we ran a set of ablations on the graph masking strategy as well as on the effect of mixing fully and partially masked training examples. To this end, we trained the 28-million parameter OmniInvent variant on the ChEMBL-derived training dataset using RDKit randomization and evaluated performance on substructure-constrained direct sampling. Results may be found in Table 4, while details on the experiments may be found in Section I. We find that a naïve span masking scheme, where training examples are formed

by randomly masking out spans in SMILES sequences, results in a noticeable decrease in validity and uniqueness. As care was taken to ensure that the distributions of the two cases were identical in terms of the number of masks and lengths of each span, this implies that aligning the masks to the underlying graph structure during pretraining is important for downstream task performance. We additionally find that pretraining only on partially masked SMILES, i.e., with $p_{\text{full}} = 0$, does not enhance performance compared to OmniInvent; our pretraining strategy does thus not appear to be detrimental to the model's capability of conducting substructure-constrained design.

Table 4: Results from substructure constrained generation on fragments from ten known drugs (Noutahi et al., 2024) for the ablations. Results were averaged over five independent runs, with errors representing the standard deviation.

Ablation	Task	Valid. (%)	Uniq. (%)	Diversity	Distance
OmniInvent	Fragm. link.	98.6 ± 0.6	80.6 ± 1.5	0.541 ± 0.001	0.500 ± 0.002
	Scaffold dec.	98.4 ± 0.4	88.6 ± 1.2	0.535 ± 0.004	0.453 ± 0.004
	Motif ext.	98.9 ± 0.2	93.1 ± 0.7	0.651 ± 0.003	0.631 ± 0.003
Masking ablation	Fragm. link.	95.9 ± 0.6	66.4 ± 0.8	0.516 ± 0.003	0.504 ± 0.002
	Scaffold dec.	89.9 ± 0.9	83.7 ± 0.6	0.519 ± 0.004	0.439 ± 0.004
	Motif ext.	96.1 ± 0.5	86.7 ± 0.8	0.631 ± 0.002	0.642 ± 0.002
$p_{\text{full}} = 0$	Fragm. link.	99.2 ± 0.3	79.4 ± 0.9	0.537 ± 0.004	0.490 ± 0.004
	Scaffold dec.	98.0 ± 0.5	85.8 ± 1.3	0.532 ± 0.001	0.461 ± 0.004
	Motif ext.	99.0 ± 0.2	92.0 ± 0.4	0.639 ± 0.003	0.628 ± 0.002

6 Conclusion

In this work, we have presented OmniInvent, a generalist CLM for molecular generation that supports both de novo and substructure-constrained design within a single framework. To our knowledge, OmniInvent is among the first multitask molecular generators operating directly on unmodified SMILES sequences and the first example of autoregressive blank infilling applied to chemistry. Its autoregressive nature additionally provides exact log-likelihoods, making it compatible with RL frameworks such as REINVENT4 for goal-directed design. Despite its general-purpose pretraining, it achieves performance comparable to both generalist and specialized models, with high validity, uniqueness, and diversity in direct sampling as well as surpassing or matching state-of-the-art results on the PMO benchmark for goal-directed de novo design. Notably, OmniInvent achieves performance comparable to specialized models for goal-directed fragment linking and scaffold decoration, despite the absence of task-specific pretraining, highlighting the versatility of the approach. The task-agnostic nature of OmniInvent's pretraining scheme also suggests that the model could, in principle, be applicable to other substructure-constrained design settings beyond fragment linking, scaffold decoration, and motif extension.

7 Limitations

While our results and ablations show that our proposed method yields capable models for molecular design, our results are not without limitations. Firstly, we did not conduct experiments on the optimal fraction of fully masked training examples (p_{full}) due to computational constraints; a different choice could in principle strike a better balance between the model's de novo and substructure-constrained design capabilities. Second, our proposed graph-masking strategy is not necessarily optimal. Replacing it with e.g. a reaction-based slicing scheme, as used by several of our baselines, would allow for better alignment between pretraining and downstream tasks; we leave this to future work, however. This may further improve performance as well as pretraining data efficiency. Thirdly, the oracle functions used in our experiments are proxies that do not necessarily reflect the complexity of real drug design tasks; this flaw is shared with our baselines, however.

AI USE STATEMENT

The authors used large language models to assist with copy-editing and to improve the clarity, grammar, and flow of the manuscript. The models did not contribute to the conception of the work, nor did they contribute to experimental design, implementation, analysis, or conclusions. The authors take full responsibility for the accuracy of the manuscript's content.

ETHICS STATEMENT

This work introduces a generative model for molecular design and does not involve human subjects, personally identifiable information, or animal experimentation. All training and evaluation data are taken from publicly available chemical databases used in accordance with their respective licenses.

We recognize that generative models for molecular design are dual-use: methods that generate molecules with desirable therapeutic properties could, in principle, also be used to generate harmful compounds. OmniInvent is designed and evaluated for drug discovery applications, and its output is controlled by user-specified scoring functions rather than any intrinsic objective. The molecules it generates are hypotheses that require extensive experimental validation and established safety, regulatory, and ethical review before any real-world use; they are not validated drug candidates. We believe the benefits of open, reproducible research on molecular generators outweigh the risks of potential misuse, particularly as the underlying methodology builds upon methods that are publicly available.

REPRODUCIBILITY STATEMENT

We provide the information necessary to reproduce our results throughout the paper and appendix. The model architecture and its modifications (prefix attention and two-dimensional positional encodings) are described in Section A and illustrated in Figure 2, with hyperparameters detailed in Section C. The data sources and preprocessing pipeline are detailed in Section B, and the on-the-fly graph-based masking as well as randomization strategies used to construct training examples are described in Section D.1. Complete training hyperparameters and optimization settings are reported in Section D.2. Evaluation protocols, including metric definitions and the direct-sampling and goal-directed setups, are given in the Methods section and expanded upon in Sections E and F.

To further support reproducibility, our (anonymized) code may be found at `https://anonymous.4open.science/r/omniinvent`. Models and pretraining data will be made public upon eventual acceptance.

REFERENCES

Josep Arús-Pous, Simon Viet Johansson, Oleksii Prykhodko, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen, and Ola Engkvist. Randomized SMILES strings improve the quality of molecular generative models. *J. Cheminform.*, 11(1):71, November 2019. doi: 10.1186/s13321-019-0393-0.

Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. Efficient Training of Language Models to Fill in the Middle, July 2022. arXiv preprint. doi: 10.48550/arXiv.2207.14255.

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. *J. Chem. Inf. Model.*, 60(12):5918–5922, December 2020. doi: 10.1021/acs.jcim.0c00915.

Nathan Brown, Marco Fiscato, Marwin H.S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking Models for de Novo Molecular Design. *J. Chem. Inf. Model.*, 59(3):1096–1108, March 2019. doi: 10.1021/acs.jcim.8b00839.

Aaron Defazio, Xingyu Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The Road Less Scheduled. In *Advances in Neural Information Processing Systems*, volume 37, pp. 9974–10007. Curran Associates, Inc., December 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/136b9a13861308c8948cd308ccd02658-Paper-Conference.pdf.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pp. 4171–4186. Association for Computational Linguistics, June 2019. doi: 10.18653/v1/N19-1423.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified Language Model Pre-training for Natural Language Understanding and Generation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/c20bb2d9a50d5ac1f713f8b34d9aac5a-Paper.pdf.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *Proceedings of the the 60th Annual Meeting of the Association for Computational Linguistics*, volume 1, pp. 320–335. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.26.

Vendy Fialková, Jiaxi Zhao, Kostas Papadopoulos, Ola Engkvist, Esben Jannik Bjerrum, Thierry Kogej, and Atanas Patronov. LibINVENT: Reaction-based Generative Scaffold Decoration for in Silico Library Design. *J. Chem. Inf. Model.*, 62(9):2046–2063, May 2022. doi: 10.1021/acs.jcim.1c00469.

Daniel Fried, Armen Aghajanyan, Jessy Lin, Sida Wang, Eric Wallace, Freda Shi, Ruiqi Zhong, Scott Yih, Luke Zettlemoyer, and Mike Lewis. Incoder: A generative model for code infilling and synthesis. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=hQwb-lbM6EL.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor Coley. Sample Efficiency Matters: A Benchmark for Practical Molecular Optimization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21342–21357. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8644353f7d307baaf29bc1e56fe8e0ec-Paper-Datasets_and_Benchmarks.pdf.

Anna Gaulton, Louisa J. Bellis, A. Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, and John P. Overington. ChEMBL: A large-scale bioactivity database for drug discovery. *Nucleic Acids Res.*, 40 (D1):D1100–D1107, January 2012. doi: 10.1093/nar/gkr777.

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 Cent. Sci.*, 4(2):268–276, February 2018. doi: 10.1021/acscentsci.7b00572.

Francesca Grisoni. Chemical language models for de novo drug design: Challenges and opportunities. *Curr. Opin. Struct. Biol.*, 79:102527, April 2023. doi: 10.1016/j.sbi.2023.102527.

Jeff Guo, Franziska Knuth, Christian Margreitter, Jon Paul Janet, Kostas Papadopoulos, Ola Engkvist, and Atanas Patronov. Link-INVENT: Generative linker design with reinforcement learning. *Digit. Discov.*, 2(2):392–408, April 2023. doi: 10.1039/D2DD00115B.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *Computer Vision – ECCV 2024*, pp. 289–305. Springer Nature Switzerland, 2025. doi: 10.1007/978-3-031-72684-2_17.

Emiel Hoogeboom, Víctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3D. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8867–8887. PMLR, 2022. URL https://proceedings.mlr.press/v162/hoogeboom22a.html.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, December 2021. URL https://openreview.net/forum?id=8nvgnORnoWr.

John J. Irwin, Teague Sterling, Michael M. Mysinger, Erin S. Bolstad, and Ryan G. Coleman. ZINC: A Free Tool to Discover Chemistry for Biology. *J. Chem. Inf. Model.*, 52(7):1757–1768, July 2012. doi: 10.1021/ci3001277.

Ross Irwin, Alessandro Tibo, Jon Paul Janet, and Simon Olsson. Semlaflow – efficient 3d molecular generation with latent attention and equivariant flow matching. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025. URL https://openreview.net/forum?id=bee2G6pEh0.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2323–2332. PMLR, 2018. URL https://proceedings.mlr.press/v80/jin18a.html.

Benno Kaech, Luis Wyss, Karsten Borgwardt, and Gianvito Grasso. Refine drugs, don't complete them: Uniform-source discrete flows for fragment-based drug discovery. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Qdu92a5DiM.

Hyeonah Kim, Minsu Kim, Sanghyeok Choi, and Jinkyoo Park. Genetic-guided GFlowNets for Sample Efficient Molecular Optimization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 42618–42648. Curran Associates, Inc., December 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/4b25c000967af9036fb9b207b198a626-Paper-Conference.pdf.

Sunghwan Kim, Paul A. Thiessen, Evan E. Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A. Shoemaker, Jiyao Wang, Bo Yu, Jian Zhang, and Stephen H. Bryant. PubChem Substance and Compound databases. *Nucleic Acids Res.*, 44(D1), January 2016. doi: 10.1093/nar/gkv951. URL https://doi.org/10.1093/nar/gkv951.

Fabian P. Krüger, Nicklas Österbacka, Mikhail Kabeshov, Ola Engkvist, and Igor Tetko. MolEncoder: Towards optimal masked language modeling for molecules. *Digit. Discov.*, 4(12):3552–3566, December 2025. doi: 10.1039/D5DD00369E.

Maxime Langevin, Hervé Minoux, Maximilien Levesque, and Marc Bianciotto. Scaffold-Constrained Molecular Generation. *J. Chem. Inf. Model.*, 60(12):5637–5646, December 2020. doi: 10.1021/acs.jcim.0c01015.

Seul Lee, Karsten Kreis, Srimukh Prasad Veccham, Meng Liu, Danny Reidenbach, Saee Gopal Paliwal, Arash Vahdat, and Weili Nie. Molecule Generation with Fragment Retrieval Augmentation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 132463–132490. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ef107bb91dae96368c856b1064370bd0-Paper-Conference.pdf.

Seul Lee, Karsten Kreis, Srimukh Prasad Veccham, Meng Liu, Danny Reidenbach, Yuxing Peng, Saee Gopal Paliwal, Weili Nie, and Arash Vahdat. GenMol: A Drug Discovery Generalist with Discrete Diffusion. In *Proceedings of the 42nd International Conference on Machine Learning*,

volume 267, pp. 33205–33226. PMLR, June 2025. URL https://proceedings.mlr.press/v267/lee25o.html.

Hannes H. Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H. Mervin, and Ola Engkvist. Reinvent 4: Modern AI–driven generative molecule design. *J. Cheminform.*, 16(1):20, February 2024. doi: 10.1186/s13321-024-00812-5.

Soma Mandal, Mee'nal Moudgil, and Sanat K. Mandal. Rational drug design. *Eur. J. Pharmacol.*, 625(1):90–100, December 2009. doi: 10.1016/j.ejphar.2009.06.065.

Borja Medina, Alessandro Tibo, Jiazhen He, Jon Paul Janet, and Nicklas Österbacka. Diversity Beats Size Scaling for Chemical Language Models. *npj Drug Discov.*, 3(1):26, August 2026. doi: 10.1038/s44386-026-00056-3.

Rocío Mercado, Tobias Rastemo, Edvard Lindelöf, Günter Klambauer, Ola Engkvist, Hongming Chen, and Esben Jannik Bjerrum. Graph networks for molecular design. *Mach. Learn.: Sci. Technol.*, 2(2):025023, March 2021. doi: 10.1088/2632-2153/abcf91.

Christophe Muller, Obdulia Rabal, and Constantino Diaz Gonzalez. Artificial Intelligence, Machine Learning, and Deep Learning in Real-Life Drug Design Cases. In Alexander Heifetz (ed.), *Artificial Intelligence in Drug Design*, pp. 383–407. Springer US, New York, NY, 2022. ISBN 978-1-0716-1787-8. doi: 10.1007/978-1-0716-1787-8_16.

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

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020. URL https://dl.acm.org/doi/10.5555/3455716.3455856.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and Effective Masked Diffusion Language Models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/eb0b13cc515724ab8015bc978fdde0ad-Paper-Conference.pdf.

Petra Schneider, W. Patrick Walters, Alleyn T. Plowright, Norman Sieroka, Jennifer Listgarten, Robert A. Goodnow, Jasmin Fisher, Johanna M. Jansen, José S. Duca, Thomas S. Rush, Matthias Zentgraf, John Edward Hill, Elizabeth Krutoholow, Matthias Kohler, Jeff Blaney, Kimito Funatsu, Chris Luebkemann, and Gisbert Schneider. Rethinking drug design in the artificial intelligence era. *Nat. Rev. Drug Discov.*, 19(5):353–364, May 2020. doi: 10.1038/s41573-019-0050-3.

Philippe Schwaller, Théophile Gaudin, Dávid Lányi, Costas Bekas, and Teodoro Laino. "Found in Translation": Predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chem. Sci.*, 9(28):6091–6098, 2018. doi: 10.1039/C8SC02339E.

Tianxiao Shen, Hao Peng, Ruoqi Shen, Yao Fu, Zaid Harchaoui, and Yejin Choi. FiLM: Fill-in Language Models for Any-Order Generation, October 2023. arXiv preprint. doi: 10.48550/arXiv.2310.09930.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, February 2024. doi: 10.1016/j.neucom.2023.127063.

Morgan Thomas, Mazen Ahmad, Gary Tresadern, and Gianni de Fabritiis. PromptSMILES: Prompting for scaffold decoration and fragment linking in chemical language models. *J. Cheminform.*, 16(1):77, July 2024. doi: 10.1186/s13321-024-00866-5.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models, February 2023. arXiv preprint. doi: 10.48550/arXiv.2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36, February 1988. doi: 10.1021/ci00057a005.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-demos.6.

A LLaMA MODIFICATIONS

OmniInvent is based on the LLaMA architecture (Touvron et al., 2023). Following the GLM approach by Du et al. (2022), we apply two modifications: we apply a prefix attention mask (Dong et al., 2019; Raffel et al., 2020) rather than a causal one and utilize a two-dimensional RoPE variant (Su et al., 2024) based on intermask and intramask positions as shown in Figures 2b) and 5. These are described in detail below.

A.1 PREFIX ATTENTION

The attention mechanism is the core operation in transformer models, allowing each token representation to be updated based on information from other tokens in the sequence (Vaswani et al., 2017). By learning which tokens are most relevant to each position, attention enables the model to capture contextual dependencies that support both sequence understanding and token generation. Figure 4 provides an overview of the different attention masks used for transformers.

With a bidirectional attention mask, as used in encoder transformers such as BERT (Devlin et al., 2019), each token attends to all tokens in the sequence, enabling the model to learn contextualized representations that can be used for downstream tasks. In contrast, the causal attention mask of decoder transformers such as LLaMA (Touvron et al., 2023) allows each token to attend only to itself and preceding tokens, enabling the generation of new tokens conditioned on an optional input as well as previously generated tokens.

A prefix attention mask (Dong et al., 2019; Raffel et al., 2020) combines properties of bidirectional and causal attention to enable autoregressive generation of a suffix conditioned on a prefix, i.e., a given input sequence, as illustrated in Figure 2a) as well as 4. Tokens in the prefix attend to all other prefix tokens, while generated tokens attend to the prefix as well as preceding suffix tokens. In this work, the prefix is a partially masked SMILES sequence, or a single mask token, and the model is trained to generate the missing segments.

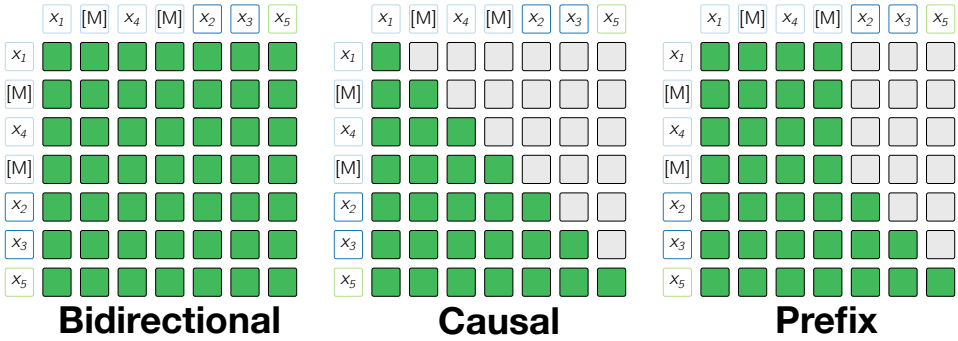


Figure 4: An illustration of attention masking patterns used in transformer models applied to the token sequence $(x_1, [M], x_4, [M], x_2, x_3, x_5)$. Bidirectional attention, used in encoder models, allows for all tokens to attend to each other. Causal attention, used in decoder models, allows tokens to attend only to previous tokens in the sequence. Prefix attention combines these two types: bidirectional attention is applied to the prefix (the first four tokens in the example above), while causal attention is applied to the later tokens in the sequence.

A.2 TWO-DIMENSIONAL POSITIONAL EMBEDDINGS

The transformer attention mechanism has no inherent awareness of token positions within a sequence. Positional embeddings are therefore typically introduced to provide this information. The LLaMA architecture uses the RoPE scheme introduced by Su et al. (2024), which encodes positional information as rotations of token-level hidden states. For standard language modeling, a one-dimensional embedding scheme is generally sufficient as text is inherently sequential. In the GLM approach, the embedding instead encodes both the positions of mask tokens in the prefix (*intermask*) and the sequential positions of the tokens within each masked segment (*intramask*). To accomplish this, we utilize a two-dimensional positional embedding scheme as illustrated in Fig.

5. The approach, often called axial 2D RoPE, has been applied to vision transformers (Heo et al., 2025) but was proposed by Su[1] as well as Biderman *et al.*[2] in blog posts as early as 2021. Conceptually, separate one-dimensional RoPEs are applied to the intermask and intramask positions, thus encoding both dimensions simultaneously.

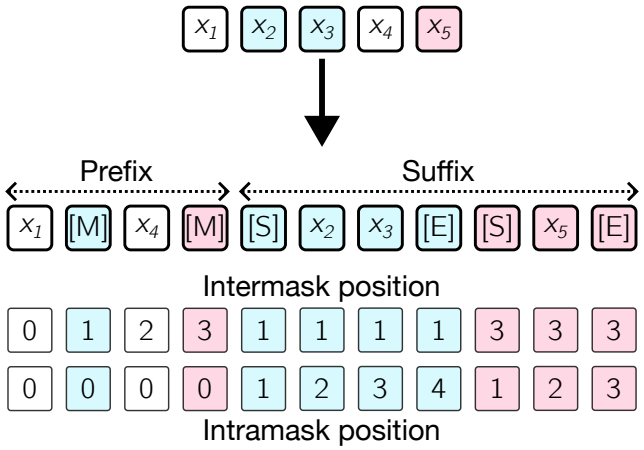


Figure 5: An illustration of the two-dimensional position embedding scheme used in this work. Two sections of the initial token sequence are replaced with mask tokens ([M]). The masked-out sections are put between start ([S]) and end ([E]) tokens and moved to the end of the sequence to form the suffix. Two positions are recorded for each token: the intermask position and the intramask position. The intermask position represents sequential positions in the prefix as well as the prefix position of the mask token corresponding to each part of the suffix. The intramask position is zero for the entire prefix and represents sequential positions within each part of the suffix.

Following Su et al. (2024), the output of the self-attention mechanism for a token at position m in a sequence may be expressed as

$$\text{Att.}_m = \frac{\sum_{n=1}^{N} \exp(\mathbf{q}_m^\top \mathbf{k}_n / \sqrt{d}) \mathbf{v}_n}{\sum_{j=1}^{N} \exp(\mathbf{q}_m^\top \mathbf{k}_j / \sqrt{d})}, \tag{1}$$

where $\mathbf{q}_i = \mathbf{W}_q \mathbf{x}_i$, $\mathbf{k}_i = \mathbf{W}_k \mathbf{x}_i$, and $\mathbf{v}_i = \mathbf{W}_v \mathbf{x}_i$ are the query, key, and value vectors resulting from multiplying the corresponding weight matrix $\mathbf{W}_{\{q,k,v\}}$ with the d-dimensional internal representation of token $\mathbf{x}_i$ at position i. Without an explicit way of encoding position, self-attention is agnostic to token order; a positional embedding may be utilized to rectify this. As proposed by Su et al. (2024), RoPE accomplishes this by rotating the key and query vectors of each token based on the relative positions of token pairs. For one-dimensional data such as language, this is done using the rotation matrix

$$\mathbf{R}^d_{\Theta,m} = \begin{pmatrix} \cos m\theta_1 & -\sin m\theta_1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \sin m\theta_1 & \cos m\theta_1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos m\theta_2 & -\sin m\theta_2 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \sin m\theta_2 & \cos m\theta_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/2-1} & -\sin m\theta_{d/2-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/2-1} & \cos m\theta_{d/2-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cos m\theta_{d/2} & -\sin m\theta_{d/2} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \sin m\theta_{d/2} & \cos m\theta_{d/2} \end{pmatrix}, \tag{2}$$

where $\Theta = \{\theta_i = 10000^{-2(i-1)/d}, i \in [1, 2, \ldots, d/2]\}$ is a pre-defined set of rotational angles, d the attention head dimension, and m the sequence position. The key and query weight vectors are then rotated using

$$\mathbf{q}_m^\top \mathbf{k}_n = \left(\mathbf{R}^d_{\Theta,m} \mathbf{W}_q \mathbf{x}_m\right)^\top \left(\mathbf{R}^d_{\Theta,n} \mathbf{W}_k \mathbf{x}_n\right) = \mathbf{x}_m^\top \mathbf{W}_q^\top \mathbf{R}^d_{\Theta,n-m} \mathbf{W}_k \mathbf{x}_n, \tag{3}$$

[1] https://spaces.ac.cn/archives/8397

[2] https://blog.eleuther.ai/rotary-embeddings/

where $\mathbf{R}^d_{\Theta,n-m} = (\mathbf{R}^d_{\Theta,m})^\top \mathbf{R}^d_{\Theta,n}$. The effective rotation thus depends on $n - m$, i.e., the relative distance between two tokens. This can be extended to two-dimensional positional data (e.g. images, or the intermask and intramask positions in this work) by applying separate one-dimensional RoPE along each dimensional axis. The rotation matrix above then becomes

$$\mathbf{R}^d_{\Theta,m,l} = \begin{pmatrix} \cos m\theta_1 & -\sin m\theta_1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \sin m\theta_1 & \cos m\theta_1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos l\theta_1 & -\sin l\theta_1 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \sin l\theta_1 & \cos l\theta_1 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/4} & -\sin m\theta_{d/4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/4} & \cos m\theta_{d/4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cos l\theta_{d/4} & -\sin l\theta_{d/4} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \sin l\theta_{d/4} & \cos l\theta_{d/4} \end{pmatrix}, \quad (4)$$

where m and l are the positions of a given token along the first and second dimensional axes, respectively.

B DATA PREPROCESSING

SMILES sequences are standardized following the same steps as the REINVENT4 data preprocessing pipeline (Loeffler et al., 2024):

- Isotope information is removed
- For SMILES sequences encoding several molecules/fragments, only the largest organic molecule/fragment is kept
- Molecules are normalized, i.e., non-standard functional groups are corrected and charges recombined
- Molecules are neutralized
- Explicit hydrogen atoms are removed
- Molecules are re-ionized such that strong acids ionize first

A filtering step is applied after standardization, where any molecules not adhering to

- Num. carbon atoms ≥ 2
- $2 \leq$ Num. heavy atoms ≤ 90
- Molecular weight ≤ 1200 Da
- Max. num. rings ≤ 12
- Max. ring size ≤ 9 atoms
- No atom types outside `[C, O, N, F, S, Cl, Br, P, I]`

are discarded.

The vocabulary used by the models we train is formed by applying the tokenizer of Schwaller et al. (2018) to the preprocessed ChEMBL 35 database. Any atomic tokens occurring in fewer than 0.01 % of all molecules in the pre-processed database are removed from the vocabulary, and any SMILES containing tokens not in the final vocabulary are removed from the dataset. Finally, the ring tokens %10, %11, and %12 are removed from the vocabulary, resulting in a vocabulary size of 47 excluding special tokens.

The pre-processing pipeline above is applied to PubChem Compound and ZINC 250K as well; the same ChEMBL 35-derived vocabulary is used in those cases as well. The PubChem Compound database is continually updated; we downloaded a snapshot on 2026-01-27 at 15:50 Central European Time. The ZINC 250K database was downloaded from the GitHub repository for the Practical Molecular Optimization benchmark.[3]

[3] `https://github.com/wenhao-gao/mol_opt`

C MODEL ARCHITECTURE

Table 5 shows the hyperparameter choices for OmniInvent and OmniInvent$_{\mathrm{L}}$. Apart from the use of 2D RoPE and prefix attention, other settings are the same as those of LLaMA (Touvron et al., 2023): bias-free attention and feed-forward layers, SiLU activation functions, equal number of attention and key/value heads, i.e., standard multi-head attention, as well as $\theta = 10\,000$ for RoPE. Model weight initialization is done using the default settings of the `transformers` library LLaMA implementation.

Table 5: Model hyperparameters for OmniInvent and OmniInvent$_{\mathrm{L}}$.

Parameter	OmniInvent	OmniInvent$_{\mathrm{L}}$
Layers	12	16
Heads	6	8
Feed-forward dim.	1 536	2 048
Head dim.	64	64
Embedding/Hidden dim.	384	512

D PRETRAINING

D.1 ON-THE-FLY GRAPH MASKING

A given SMILES sequence requires processing to be turned into an OmniInvent training example. For each molecule, we either mask the full SMILES or generate a partial mask comprising 1–4 disjoint connected subgraphs mapped to the SMILES sequence. The former case is chosen with probability p_{full} to capture de novo design scenarios. The latter case, which accounts for the majority of training examples, captures substructure-constrained design. Both the number of subgraphs to be masked and the total number of masked atoms are randomized, with equal probability for each possibility. The mask is initialized by selecting random non-adjacent seed atoms, one for each subgraph, and then expanded by repeatedly sampling atoms adjacent to the current subgraphs with equal probability until the target number of masked atoms is reached. Subgraph growth is constrained so that they remain disjoint. The selected atoms are then mapped onto a randomized SMILES representation of the molecule and the corresponding SMILES subsequences are masked.

Because not all SMILES tokens correspond directly to atoms, additional processing is required. Branches are represented by parentheses, ring closures by digits, and certain bond types by dedicated symbols. As these tokens are not selected explicitly by the graph-based masking procedure, simple consistency rules are applied during masking. Parentheses are masked when all atoms within the branch as well as its root atom are masked. Bond symbols are masked when an adjacent atomic symbol is masked. Ring symbols are masked when the atomic symbol to their left is masked, i.e., when the masked atom opens or closes the ring. Consecutive masked tokens are then collapsed into a single `[MASK]` token. The resulting partially masked SMILES sequence constitutes the prefix. The suffix consists of the masked token spans corresponding to each `[MASK]`, with start-of-sequence and end-of-sequence tokens added to the beginning and end of each span. Concatenation of the prefix and suffix yields the final training example.

D.2 TRAINING SETTINGS

Pretraining was conducted using standard cross-entropy loss applied to the suffix under teacher forcing. The schedule-free AdamW variant introduced by Defazio et al. (2024) with a learning rate of $5 \cdot 10^{-3}$, weight decay of 0.02, $\beta_1 = 0.9$, and $\beta_2 = 0.98$ was used for optimization. Training was performed on 8 GPUs with a per-GPU batch size of 128. Dropout with probability 0.1 was applied to the attention layers. The ChEMBL-pretrained models were trained for 600 epochs with $p_{\text{full}} = 0.025$, thus producing fully masked versions of the entire dataset 15 times on average, while the PubChem-pretrained models were trained for 12 epochs with $p_{\text{full}} = 0.075$, thus producing full masks for 90 % of the dataset on average. The epoch counts were chosen such that the two cases are roughly matched in terms of compute. The learning rate was warmed up over the first epoch.

E SUBSTRUCTURE-CONSTRAINED INPUT GENERATION

In substructure-constrained settings, OmniInvent uses partially masked SMILES sequences representing input fragments. As a given molecule may have several equivalent SMILES representations, the masked input SMILES may also have several representations. Consider the two fragments `C*` and `*c1ccccc1`, with the asterisks here representing attachment points; RDKit parses this symbol as a wildcard, i.e., an atom with unknown identity. To construct an OmniInvent input, these are first merged into one SMILES sequence, with additional wildcard symbols to allow randomization (`C****c1ccccc1`). The result is parsed by RDKit, after which new SMILES representations can be generated through permutation or RDKit randomization. All adjacent wildcard symbols are then replaced with `[MASK]` tokens, yielding the final input. The additional wildcard tokens ensure that SMILES such as `C****c1ccccc1`, `*(C)***c1ccccc1`, and `*(*c1ccccc1)**C` are accessible, thus creating masked inputs with a different number of masking tokens.

For the substructure-constrained direct sampling experiments, we generate 10000 randomized SMILES and insert mask tokens in each independent run. We then remove duplicates and randomly pick 10 of these, without replacement, to use as model inputs, generating 10 outputs for each for a total of 100 samples.

F GOAL-DIRECTED DESIGN SETUPS

Details on the RL parameters and scoring objectives are shown below. The Adam optimizer was used to update model parameters in these experiments, with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

F.1 DE NOVO DESIGN

Table 6 shows the hyperparameter settings used for the Practical Molecular Optimization benchmark runs with OmniInvent; they match those used for REINVENT$_{\text{PMO}}$ by Gao et al. (2022). Patience refers to the number of epochs allowed before early stopping; the run ends prematurely if the top-100 scores do not improve for 5 consecutive epochs. The replay buffer size is the number of high-scoring molecules to be stored in memory, while the replay memory size reflects how many of those are fed through the model each epoch. REINVENT4 supports setting the former to a higher number than the latter, thus replaying only a portion of the SMILES saved in memory.

For the prescreening experiments we set the patience to 100 instead to avoid severe under-utilization of the oracle budget.

Table 6: Hyperparameter settings for OmniInvent as well as REINVENT$_4$ for the PMO experiments

Parameter	Value
Batch size	64
Learning rate	$5 \cdot 10^{-4}$
σ	500
Patience	5
Replay buffer size	24
Replay memory size	24

F.2 SUBSTRUCTURE-CONSTRAINED DESIGN

We use the same hyperparameters for all of the substructure-constrained design experiments, shown in Table 7. We apply either RDKit or permutation randomization once per epoch to the input fragments during the goal-directed substructure-constrained design experiments; 64 samples are thus drawn for a single input in each epoch.

Table 7: Hyperparameter settings for the scaffold decoration and fragment linking experiments.

Parameter	Value
Epochs	200
Batch size	64
Learning rate	10^{-4}
σ	120

F.2.1 SCAFFOLD DECORATION

In the scaffold decoration experiment, the goal is to decorate the scaffold shown in Figure 6 while optimizing for DRD2 activity predicted by a machine learning model developed by Olivecrona et al. (2017) as implemented in the Therapeutics Data Commons package (Huang et al., 2021). The score is additionally set to zero if certain substructures occur in the resulting molecule. This is formulated as a set of SMARTS patterns, shown in Table 8.

F.2.2 FRAGMENT LINKING

In the fragment linking experiments, the task is to generate linkers for the fragments shown in Figure 7 while adhering to a set of physicochemical properties as well as a linker length range. Table 9 shows the physicochemical properties that were optimized for. An additional component enforced linker linearity; we performed reinforcement learning both with and without this constraint. Experiments were conducted with four different target linker length ranges: 4–6 atoms, 7–9 atoms,

Table 8: SMARTS patterns used for substructure avoidance in the scaffold decoration experiment.

`[*;r8]`	`[*;r9]`	`[*;r10]`	`[*;r11]`
`[*;r12]`	`[*;r13]`	`[*;r14]`	`[*;r15]`
`[*;r16]`	`[*;r17]`	`[#8][#8]`	`[#6;+]`
`[#16][#16]`	`C(=[O,S])[O,S]`	`[#7;!n][#7;!n]`	`C#C`

`[#7;!n][S;!$(S(=O)=O)]`

`[#7;!n][C;!$(C(=[O,N])[N,O])][#16;!s]`

`[#7;!n][C;!$(C(=[O,N])[N,O])][#7;!n]`

`[#7;!n][C;!$(C(=[O,N])[N,O])][#8;!o]`

`[#8;!o][C;!$(C(=[O,N])[N,O])][#16;!s]`

`[#8;!o][C;!$(C(=[O,N])[N,O])][#8;!o]`

`[#16;!s][C;!$(C(=[O,N])[N,O])][#16;!s]`

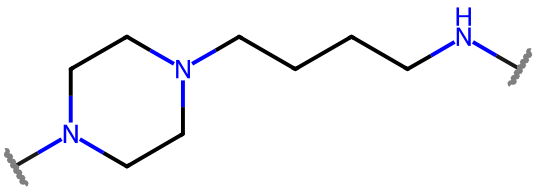


Figure 6: The scaffold to be decorated in the scaffold decoration experiments. The attachment points are shown in grey.

10–12 atoms, 13–15 atoms, as well as a baseline without any target length. Property values were transformed into score values in the range 0–1 using a double sigmoid function, ensuring that the score quickly decays to 0 as the property deviates from the target range. The total score is the geometric mean of all scoring components.

Table 9: Physicochemical property target values used for the fragment linking experiments.

Property	Values
TPSA	$< 250\,\text{Å}^2$
SlogP	3.5–6.0
H-bond acceptors	≤ 16
H-bond donors	3–6
Rotatable bonds	< 25

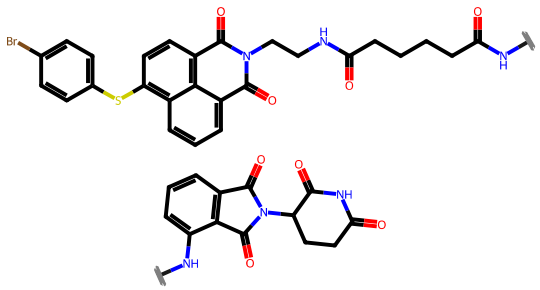


Figure 7: The two fragments to be linked in the fragment linking experiments. Attachment points are shown as grey lines.

G SAMPLES FROM THE MODEL

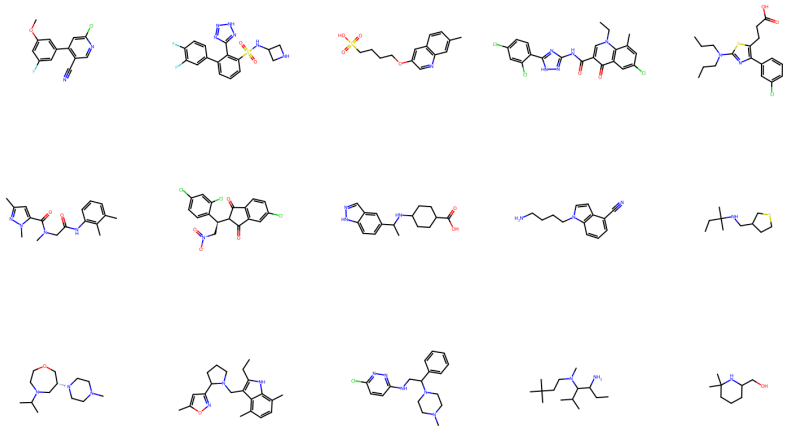
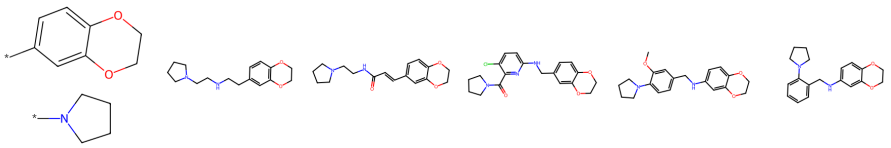
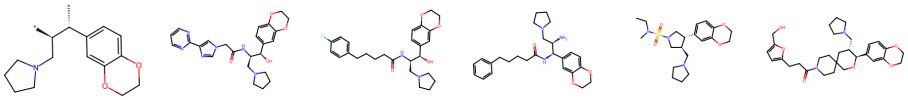


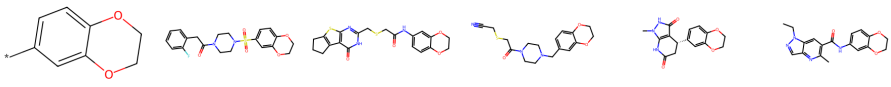
Figure 8: 15 randomly drawn samples from the OmniInvent de novo generation experiment shown in Table 1.



(a) Fragment linking.



(b) Scaffold decoration.



(c) Motif extension.

Figure 9: Five samples drawn from the OmniInvent substructure-constrained experiment shown in Table 2 for each Eliglustat-derived input. The leftmost molecule in each subfigure represents the task-specific input.

H ADDITIONAL RESULTS

H.1 DE NOVO DIRECT SAMPLING

Model training was conducted using SMILES derived from RDKit as well as permutation randomization on both PubChem and ChEMBL. De novo direct sampling results for OmniInvent as well as OmniInvent$_{\mathrm{L}}$ pretrained on ChEMBL and PubChem using RDKit as well as permutation randomization are shown in Table 10.

Table 10: De novo direct sampling results for OmniInvent and OmniInvent$_{\mathrm{L}}$ pretrained on PubChem and ChEMBL with training-time RDKit as well as permutation randomization. The names of models included in the main manuscript are bolded. Results were averaged over three independent runs, with errors representing the standard deviation.

	Model	Validity (%)	Uniqueness (%)	Diversity
RDKit PubChem	**OmniInvent**	99.0 ± 0.0	100.0 ± 0.0	0.895 ± 0.001
	OmniInvent$_{\mathrm{L}}$	99.2 ± 0.0	100.0 ± 0.0	0.897 ± 0.001
Permut. PubChem	OmniInvent	99.3 ± 0.0	100.0 ± 0.0	0.896 ± 0.001
	OmniInvent$_{\mathrm{L}}$	99.4 ± 0.2	100.0 ± 0.0	0.896 ± 0.000
RDKit ChEMBL	OmniInvent	98.1 ± 0.1	100.0 ± 0.0	0.888 ± 0.000
	OmniInvent$_{\mathrm{L}}$	98.8 ± 0.1	100.0 ± 0.0	0.887 ± 0.000
Permut. ChEMBL	OmniInvent	98.7 ± 0.1	100.0 ± 0.0	0.888 ± 0.001
	OmniInvent$_{\mathrm{L}}$	99.1 ± 0.1	100.0 ± 0.0	0.887 ± 0.000

H.2 SUBSTRUCTURE-CONSTRAINED DIRECT SAMPLING

The randomization procedure we use for input randomization may result in a different number of mask tokens for a given set of input fragments. To investigate the sensitivity of OmniInvent to this, we generate random inputs under two conditions: placing no restriction on the number of mask tokens of the input (unrestricted) and requiring the input to contain the maximum possible number of mask tokens. The maximum number of masking tokens for each set of input fragments was determined empirically by generating 100 000 masked SMILES. The results are shown in Tables 11, 12, 13, and 14.

Table 11: Results from substructure constrained generation on fragments from ten known drugs (Noutahi et al., 2024) for OmniInvent and OmniInvent$_L$ pretrained on PubChem and ChEMBL with RDKit randomization, placing no restriction on the number of mask tokens in the input (unr.) as well as maximizing the number of mask tokens in the input (max.). RDKit randomization was used to construct model inputs. The models in bold were included in the main manuscript. Results were averaged over five independent runs, with errors representing the standard deviation.

Training: RDKit randomization. Inputs: RDKit randomization.

	Model	Task	Valid. (%)	Uniq. (%)	Diversity	Distance
Unr. PubChem	OmniInvent	Fragm. link.	99.8 ± 0.2	50.6 ± 1.9	0.511 ± 0.006	0.498 ± 0.004
		Scaffold dec.	95.7 ± 1.4	71.4 ± 1.7	0.520 ± 0.008	0.441 ± 0.004
		Motif ext.	99.6 ± 0.2	86.6 ± 1.6	0.642 ± 0.001	0.629 ± 0.003
	OmniInvent$_\text{L}$	Fragm. link.	99.8 ± 0.2	51.6 ± 0.7	0.505 ± 0.004	0.479 ± 0.003
		Scaffold dec.	99.1 ± 0.5	66.8 ± 1.1	0.505 ± 0.006	0.422 ± 0.006
		Motif ext.	99.6 ± 0.2	90.8 ± 0.7	0.641 ± 0.001	0.614 ± 0.003
Max. PubChem	**OmniInvent**	Fragm. link.	99.5 ± 0.2	83.8 ± 1.2	0.532 ± 0.003	0.524 ± 0.001
		Scaffold dec.	97.6 ± 0.6	93.8 ± 0.9	0.568 ± 0.005	0.474 ± 0.005
		Motif ext.	99.0 ± 0.3	97.4 ± 0.4	0.664 ± 0.003	0.656 ± 0.002
	OmniInvent$_\text{L}$	Fragm. link.	99.4 ± 0.2	80.8 ± 1.1	0.530 ± 0.004	0.496 ± 0.005
		Scaffold dec.	98.0 ± 0.6	92.2 ± 1.3	0.553 ± 0.005	0.449 ± 0.006
		Motif ext.	99.3 ± 0.3	97.9 ± 0.9	0.656 ± 0.002	0.637 ± 0.001
Unr. ChEMBL	OmniInvent	Fragm. link.	99.6 ± 0.1	46.4 ± 2.4	0.512 ± 0.005	0.463 ± 0.007
		Scaffold dec.	99.8 ± 0.1	53.2 ± 1.9	0.458 ± 0.011	0.402 ± 0.006
		Motif ext.	99.6 ± 0.1	73.9 ± 2.2	0.622 ± 0.010	0.605 ± 0.008
	OmniInvent$_\text{L}$	Fragm. link.	99.6 ± 0.2	40.6 ± 2.5	0.500 ± 0.012	0.448 ± 0.009
		Scaffold dec.	98.5 ± 0.4	53.9 ± 2.1	0.461 ± 0.005	0.401 ± 0.003
		Motif ext.	99.7 ± 0.2	70.1 ± 1.5	0.612 ± 0.003	0.585 ± 0.001
Max. ChEMBL	OmniInvent	Fragm. link.	98.6 ± 0.6	80.6 ± 1.5	0.541 ± 0.001	0.500 ± 0.002
		Scaffold dec.	98.4 ± 0.4	88.6 ± 1.2	0.535 ± 0.004	0.453 ± 0.004
		Motif ext.	98.9 ± 0.2	93.1 ± 0.7	0.651 ± 0.003	0.631 ± 0.003
	OmniInvent$_\text{L}$	Fragm. link.	99.3 ± 0.3	68.5 ± 1.8	0.520 ± 0.005	0.473 ± 0.003
		Scaffold dec.	98.0 ± 0.6	81.8 ± 1.2	0.514 ± 0.006	0.440 ± 0.005
		Motif ext.	99.2 ± 0.2	87.3 ± 1.4	0.633 ± 0.002	0.616 ± 0.005

Table 12: Results from substructure constrained generation on fragments from ten known drugs (Noutahi et al., 2024) for OmniInvent and OmniInvent$_L$ pretrained on PubChem and ChEMBL with permutation randomization, placing no restriction on the number of mask tokens in the input (unr.) as well as maximizing the number of mask tokens in the input (max.). Permutation randomization was used to construct model inputs. Results were averaged over five independent runs, with errors representing the standard deviation.

Training: Permutation randomization. Inputs: Permutation randomization.

	Model	Task	Valid. (%)	Uniq. (%)	Diversity	Distance
Unr. PubChem	OmniInvent	Fragm. link.	99.8 ± 0.1	49.7 ± 1.0	0.502 ± 0.005	0.510 ± 0.004
		Scaffold dec.	99.4 ± 0.3	72.0 ± 1.5	0.502 ± 0.002	0.424 ± 0.004
		Motif ext.	99.7 ± 0.2	89.4 ± 1.1	0.644 ± 0.003	0.633 ± 0.002
	OmniInvent$_L$	Fragm. link.	100.0 ± 0.0	49.0 ± 1.2	0.491 ± 0.005	0.501 ± 0.003
		Scaffold dec.	98.3 ± 0.8	71.7 ± 0.5	0.498 ± 0.005	0.423 ± 0.005
		Motif ext.	99.7 ± 0.1	92.6 ± 1.1	0.642 ± 0.003	0.618 ± 0.002
Max. PubChem	OmniInvent	Fragm. link.	99.5 ± 0.2	80.9 ± 0.7	0.516 ± 0.004	0.526 ± 0.003
		Scaffold dec.	98.8 ± 0.5	94.0 ± 0.9	0.538 ± 0.004	0.446 ± 0.004
		Motif ext.	99.2 ± 0.3	96.8 ± 0.8	0.650 ± 0.004	0.668 ± 0.002
	OmniInvent$_L$	Fragm. link.	99.7 ± 0.1	78.5 ± 1.4	0.510 ± 0.002	0.511 ± 0.001
		Scaffold dec.	98.3 ± 0.8	90.7 ± 1.4	0.526 ± 0.007	0.436 ± 0.003
		Motif ext.	99.3 ± 0.2	96.1 ± 0.6	0.645 ± 0.002	0.658 ± 0.002
Unr. ChEMBL	OmniInvent	Fragm. link.	99.0 ± 0.3	42.8 ± 1.5	0.507 ± 0.004	0.490 ± 0.008
		Scaffold dec.	98.5 ± 0.4	58.6 ± 2.9	0.472 ± 0.013	0.422 ± 0.008
		Motif ext.	99.5 ± 0.3	72.2 ± 1.5	0.618 ± 0.005	0.606 ± 0.006
	OmniInvent$_L$	Fragm. link.	99.7 ± 0.1	39.4 ± 2.1	0.500 ± 0.007	0.474 ± 0.007
		Scaffold dec.	99.6 ± 0.2	55.3 ± 1.7	0.466 ± 0.012	0.407 ± 0.008
		Motif ext.	99.5 ± 0.4	70.2 ± 1.0	0.612 ± 0.006	0.601 ± 0.005
Max. ChEMBL	OmniInvent	Fragm. link.	98.9 ± 0.4	77.6 ± 0.8	0.527 ± 0.004	0.514 ± 0.002
		Scaffold dec.	98.0 ± 0.3	86.7 ± 1.6	0.521 ± 0.004	0.444 ± 0.006
		Motif ext.	98.9 ± 0.4	90.3 ± 1.0	0.636 ± 0.003	0.650 ± 0.003
	OmniInvent$_L$	Fragm. link.	98.9 ± 0.2	70.7 ± 2.1	0.524 ± 0.004	0.490 ± 0.003
		Scaffold dec.	98.9 ± 0.2	77.6 ± 1.6	0.504 ± 0.003	0.425 ± 0.005
		Motif ext.	99.0 ± 0.2	82.8 ± 0.7	0.619 ± 0.005	0.641 ± 0.003

Table 13: Results from substructure constrained generation on fragments from ten known drugs (Noutahi et al., 2024) for OmniInvent and OmniInvent$_L$ pretrained on PubChem and ChEMBL with RDKit randomization, placing no restriction on the number of mask tokens in the input (unr.) as well as maximizing the number of mask tokens in the input (max.). Permutation randomization was used to construct model inputs. Results were averaged over five independent runs, with errors representing the standard deviation.

Training: RDKit randomization. Inputs: Permutation randomization.

	Model	Task	Valid. (%)	Uniq. (%)	Diversity	Distance
Unr. PubChem	OmniInvent	Fragm. link.	99.8 ± 0.2	52.4 ± 2.4	0.509 ± 0.004	0.499 ± 0.003
		Scaffold dec.	95.8 ± 1.3	72.5 ± 1.6	0.519 ± 0.006	0.437 ± 0.006
		Motif ext.	99.4 ± 0.4	86.8 ± 1.3	0.642 ± 0.002	0.629 ± 0.004
	OmniInvent$_L$	Fragm. link.	99.9 ± 0.1	53.4 ± 1.6	0.507 ± 0.004	0.478 ± 0.005
		Scaffold dec.	99.4 ± 0.1	68.9 ± 2.1	0.507 ± 0.007	0.424 ± 0.006
		Motif ext.	99.5 ± 0.2	90.7 ± 0.5	0.640 ± 0.002	0.616 ± 0.002
Max. PubChem	OmniInvent	Fragm. link.	99.6 ± 0.3	83.6 ± 2.0	0.532 ± 0.003	0.524 ± 0.004
		Scaffold dec.	97.0 ± 0.6	95.1 ± 0.3	0.573 ± 0.004	0.478 ± 0.003
		Motif ext.	99.1 ± 0.2	97.3 ± 0.5	0.663 ± 0.003	0.655 ± 0.002
	OmniInvent$_L$	Fragm. link.	99.6 ± 0.2	80.2 ± 0.8	0.530 ± 0.002	0.497 ± 0.003
		Scaffold dec.	98.4 ± 0.4	92.5 ± 1.3	0.544 ± 0.005	0.441 ± 0.005
		Motif ext.	99.5 ± 0.2	98.3 ± 0.8	0.658 ± 0.005	0.635 ± 0.003
Unr. ChEMBL	OmniInvent	Fragm. link.	99.8 ± 0.2	47.0 ± 0.5	0.516 ± 0.003	0.467 ± 0.005
		Scaffold dec.	99.1 ± 0.4	56.4 ± 1.0	0.466 ± 0.005	0.408 ± 0.005
		Motif ext.	99.3 ± 0.3	74.0 ± 1.3	0.617 ± 0.006	0.602 ± 0.003
	OmniInvent$_L$	Fragm. link.	99.7 ± 0.2	39.4 ± 1.9	0.494 ± 0.012	0.434 ± 0.009
		Scaffold dec.	99.3 ± 0.4	53.2 ± 1.2	0.470 ± 0.010	0.408 ± 0.014
		Motif ext.	99.5 ± 0.2	70.1 ± 1.5	0.611 ± 0.003	0.586 ± 0.003
Max. ChEMBL	OmniInvent	Fragm. link.	99.0 ± 0.4	80.9 ± 1.2	0.540 ± 0.004	0.498 ± 0.004
		Scaffold dec.	98.2 ± 0.4	88.8 ± 0.7	0.538 ± 0.005	0.455 ± 0.006
		Motif ext.	99.0 ± 0.2	93.5 ± 0.7	0.651 ± 0.003	0.629 ± 0.002
	OmniInvent$_L$	Fragm. link.	99.2 ± 0.2	67.6 ± 1.7	0.521 ± 0.003	0.474 ± 0.001
		Scaffold dec.	97.7 ± 0.5	82.4 ± 0.8	0.517 ± 0.001	0.440 ± 0.003
		Motif ext.	98.9 ± 0.3	86.5 ± 0.8	0.631 ± 0.002	0.617 ± 0.002

Table 14: Results from substructure constrained generation on fragments from ten known drugs (Noutahi et al., 2024) for OmniInvent and OmniInvent$_L$ pretrained on PubChem and ChEMBL with permutation randomization, placing no restriction on the number of mask tokens in the input (unr.) as well as maximizing the number of mask tokens in the input (max.). RDKit randomization was used to construct model inputs. Results were averaged over five independent runs, with errors representing the standard deviation.

Training: Permutation randomization. Inputs: RDKit randomization.

	Model	Task	Valid. (%)	Uniq. (%)	Diversity	Distance
Unr. PubChem	OmniInvent	Fragm. link.	98.7 ± 0.6	49.8 ± 2.1	0.504 ± 0.007	0.510 ± 0.006
		Scaffold dec.	82.2 ± 1.6	73.6 ± 2.4	0.526 ± 0.012	0.446 ± 0.006
		Motif ext.	98.5 ± 0.7	82.7 ± 1.6	0.643 ± 0.003	0.636 ± 0.003
	OmniInvent$_\text{L}$	Fragm. link.	96.9 ± 1.2	47.0 ± 1.8	0.494 ± 0.005	0.502 ± 0.004
		Scaffold dec.	82.8 ± 2.5	76.0 ± 1.4	0.528 ± 0.002	0.440 ± 0.003
		Motif ext.	97.8 ± 1.1	85.5 ± 1.5	0.640 ± 0.002	0.619 ± 0.004
Max. PubChem	OmniInvent	Fragm. link.	96.9 ± 1.5	79.9 ± 1.6	0.513 ± 0.003	0.526 ± 0.001
		Scaffold dec.	81.7 ± 3.3	94.6 ± 0.8	0.574 ± 0.014	0.480 ± 0.012
		Motif ext.	98.4 ± 0.4	95.7 ± 0.3	0.644 ± 0.002	0.666 ± 0.002
	OmniInvent$_\text{L}$	Fragm. link.	95.9 ± 0.8	77.4 ± 1.5	0.502 ± 0.003	0.512 ± 0.002
		Scaffold dec.	85.3 ± 2.6	91.9 ± 0.8	0.555 ± 0.013	0.459 ± 0.011
		Motif ext.	96.5 ± 1.1	95.4 ± 1.1	0.644 ± 0.003	0.660 ± 0.002
Unr. ChEMBL	OmniInvent	Fragm. link.	96.1 ± 0.8	42.7 ± 2.4	0.517 ± 0.006	0.498 ± 0.005
		Scaffold dec.	79.8 ± 4.8	63.3 ± 2.5	0.503 ± 0.017	0.448 ± 0.013
		Motif ext.	97.0 ± 1.1	70.4 ± 1.6	0.621 ± 0.001	0.610 ± 0.003
	OmniInvent$_\text{L}$	Fragm. link.	95.0 ± 1.3	39.6 ± 1.3	0.496 ± 0.005	0.471 ± 0.008
		Scaffold dec.	82.5 ± 3.2	58.3 ± 1.3	0.498 ± 0.010	0.433 ± 0.008
		Motif ext.	97.1 ± 0.3	66.1 ± 1.7	0.612 ± 0.003	0.601 ± 0.004
Max. ChEMBL	OmniInvent	Fragm. link.	97.8 ± 0.3	78.3 ± 1.0	0.526 ± 0.004	0.512 ± 0.001
		Scaffold dec.	83.2 ± 1.9	89.5 ± 1.0	0.556 ± 0.009	0.474 ± 0.008
		Motif ext.	97.5 ± 0.5	89.7 ± 0.2	0.633 ± 0.003	0.651 ± 0.003
	OmniInvent$_\text{L}$	Fragm. link.	97.1 ± 1.0	70.1 ± 1.0	0.522 ± 0.004	0.488 ± 0.004
		Scaffold dec.	82.8 ± 2.3	82.0 ± 0.9	0.540 ± 0.010	0.460 ± 0.010
		Motif ext.	98.4 ± 0.2	82.1 ± 1.3	0.616 ± 0.001	0.642 ± 0.002

H.3 DE NOVO GOAL-DIRECTED DESIGN

Oracle descriptions may be found in the works of Brown et al. (2019) and Gao et al. (2022).

Table 15: Top-10 AUCs for the 23 PMO oracles for OmniInvent and OmniInvent$_{\text{L}}$ pretrained on PubChem with RDKit as well as permutation randomization. Results are reported as averages over five independent runs and the experiments were conducted **without** prescreening. Model names that are included in the main manuscript are bolded. Isom. 1 and 2 refer to isomers of `C7H8N2O2` and `C9H10N2O2PF2Cl`, respectively.

	RDKit PubChem		Permut. PubChem	
Oracle	**OmniInvent**	**OmniInvent$_{\text{L}}$**	OmniInvent	OmniInvent$_{\text{L}}$
Albuterol sim.	0.94 ± 0.01	0.93 ± 0.02	0.93 ± 0.00	0.93 ± 0.01
Amlodipine MPO	0.75 ± 0.07	0.71 ± 0.05	0.79 ± 0.03	0.74 ± 0.05
Celecoxib redisc.	0.84 ± 0.01	0.83 ± 0.01	0.83 ± 0.01	0.83 ± 0.01
Deco hop	0.93 ± 0.01	0.93 ± 0.01	0.92 ± 0.02	0.85 ± 0.12
DRD2	0.97 ± 0.01	0.97 ± 0.01	0.97 ± 0.00	0.97 ± 0.01
Fexofenadine MPO	0.85 ± 0.02	0.89 ± 0.03	0.84 ± 0.03	0.83 ± 0.03
GSK3β	0.93 ± 0.01	0.94 ± 0.00	0.93 ± 0.02	0.92 ± 0.03
Isom. 1	0.95 ± 0.02	0.96 ± 0.01	0.96 ± 0.01	0.96 ± 0.01
Isom. 2	0.84 ± 0.00	0.86 ± 0.03	0.84 ± 0.01	0.86 ± 0.05
JNK3	0.87 ± 0.02	0.91 ± 0.01	0.92 ± 0.01	0.88 ± 0.05
Median 1	0.37 ± 0.01	0.38 ± 0.02	0.37 ± 0.04	0.36 ± 0.03
Median 2	0.35 ± 0.03	0.33 ± 0.03	0.34 ± 0.02	0.32 ± 0.04
Mestranol sim.	0.93 ± 0.02	0.92 ± 0.01	0.85 ± 0.12	0.92 ± 0.01
Osimertinib MPO	0.88 ± 0.02	0.88 ± 0.02	0.87 ± 0.02	0.88 ± 0.03
Perindopril MPO	0.61 ± 0.02	0.66 ± 0.06	0.64 ± 0.08	0.64 ± 0.06
QED	0.94 ± 0.00	0.94 ± 0.00	0.94 ± 0.00	0.95 ± 0.00
Ranolazine MPO	0.84 ± 0.01	0.82 ± 0.02	0.83 ± 0.02	0.81 ± 0.01
Scaff. hop	0.70 ± 0.12	0.71 ± 0.14	0.65 ± 0.13	0.74 ± 0.12
Sitagliptin MPO	0.46 ± 0.03	0.44 ± 0.04	0.48 ± 0.13	0.50 ± 0.06
Thiothixene redisc.	0.57 ± 0.05	0.62 ± 0.06	0.65 ± 0.12	0.62 ± 0.08
Troglitazone redisc.	0.70 ± 0.09	0.73 ± 0.11	0.68 ± 0.11	0.63 ± 0.13
Valsartan SMARTS	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Zaleplon MPO	0.53 ± 0.02	0.50 ± 0.01	0.49 ± 0.02	0.52 ± 0.04
Sum	16.73 ± 0.24	16.86 ± 0.14	16.72 ± 0.22	16.64 ± 0.21

Table 16: Top-10 AUCs for the 23 PMO oracles for OmniInvent and OmniInvent$_{\text{L}}$ pretrained on ChEMBL with RDKit as well as permutation randomization. Results are reported as averages over five independent runs and the experiments were conducted **without** prescreening. Isom. 1 and 2 refer to isomers of `C7H8N2O2` and `C9H10N2O2PF2Cl`, respectively.

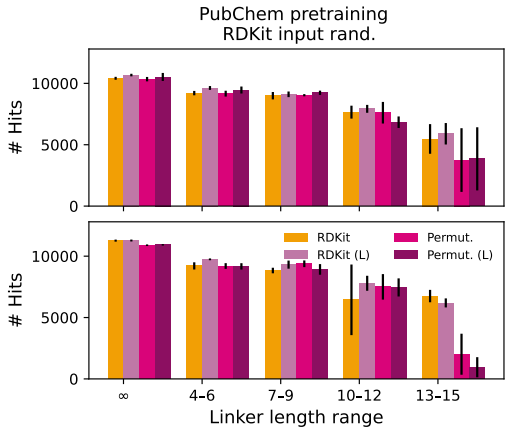
	RDKit ChEMBL		Permut. ChEMBL	
Oracle	OmniInvent	OmniInvent$_{\text{L}}$	OmniInvent	OmniInvent$_{\text{L}}$
Albuterol sim.	0.95 ± 0.01	0.94 ± 0.01	0.94 ± 0.01	0.95 ± 0.01
Amlodipine MPO	0.72 ± 0.05	0.76 ± 0.07	0.68 ± 0.05	0.69 ± 0.05
Celecoxib redisc.	0.84 ± 0.01	0.84 ± 0.00	0.85 ± 0.01	0.85 ± 0.01
Deco hop	0.94 ± 0.02	0.94 ± 0.02	0.92 ± 0.03	0.94 ± 0.04
DRD2	0.98 ± 0.00	0.98 ± 0.00	0.98 ± 0.01	0.98 ± 0.00
Fexofenadine MPO	0.85 ± 0.02	0.85 ± 0.03	0.84 ± 0.03	0.84 ± 0.02
GSK3β	0.97 ± 0.01	0.96 ± 0.01	0.96 ± 0.04	0.96 ± 0.03
Isom. 1	0.96 ± 0.01	0.96 ± 0.01	0.97 ± 0.00	0.97 ± 0.00
Isom. 2	0.82 ± 0.02	0.82 ± 0.04	0.83 ± 0.02	0.85 ± 0.01
JNK3	0.92 ± 0.02	0.93 ± 0.01	0.94 ± 0.02	0.92 ± 0.02
Median 1	0.38 ± 0.03	0.36 ± 0.01	0.37 ± 0.04	0.38 ± 0.03
Median 2	0.35 ± 0.02	0.35 ± 0.02	0.34 ± 0.04	0.32 ± 0.02
Mestranol sim.	0.94 ± 0.01	0.94 ± 0.01	0.95 ± 0.00	0.93 ± 0.03
Osimertinib MPO	0.89 ± 0.03	0.88 ± 0.01	0.88 ± 0.03	0.88 ± 0.01
Perindopril MPO	0.62 ± 0.05	0.62 ± 0.06	0.60 ± 0.02	0.61 ± 0.04
QED	0.94 ± 0.00	0.94 ± 0.00	0.94 ± 0.00	0.94 ± 0.00
Ranolazine MPO	0.83 ± 0.01	0.84 ± 0.02	0.83 ± 0.02	0.83 ± 0.02
Scaff. hop	0.66 ± 0.12	0.65 ± 0.11	0.71 ± 0.16	0.72 ± 0.15
Sitagliptin MPO	0.51 ± 0.13	0.35 ± 0.04	0.50 ± 0.09	0.40 ± 0.09
Thiothixene redisc.	0.64 ± 0.13	0.61 ± 0.03	0.66 ± 0.12	0.67 ± 0.15
Troglitazone redisc.	0.73 ± 0.07	0.70 ± 0.14	0.65 ± 0.16	0.75 ± 0.10
Valsartan SMARTS	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Zaleplon MPO	0.50 ± 0.03	0.51 ± 0.03	0.52 ± 0.04	0.49 ± 0.02
Sum	16.94 ± 0.11	16.75 ± 0.18	16.88 ± 0.28	16.87 ± 0.40

Table 17: Top-10 AUCs for the 23 PMO oracles for OmniInvent and OmniInvent$_{\text{L}}$ pretrained on PubChem with RDKit randomization. Results are reported as averages over five independent runs and the experiments were conducted **with** prescreening. Isom. 1 and 2 refer to isomers of `C7H8N2O2` and `C9H10N2O2PF2Cl`, respectively.

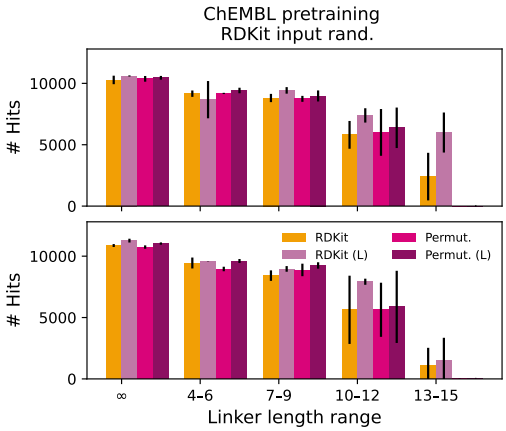
	RDKit PubChem	
Oracle	**OmniInvent**	**OmniInvent$_{\text{L}}$**
Albuterol sim.	0.98 ± 0.00	0.97 ± 0.00
Amlodipine MPO	0.87 ± 0.01	0.84 ± 0.02
Celecoxib redisc.	0.86 ± 0.01	0.86 ± 0.00
Deco hop	0.95 ± 0.01	0.96 ± 0.02
DRD2	0.98 ± 0.00	0.98 ± 0.00
Fexofenadine MPO	0.88 ± 0.02	0.89 ± 0.05
GSK3β	0.98 ± 0.00	0.97 ± 0.00
Isom. 1	0.97 ± 0.00	0.98 ± 0.00
Isom. 2	0.91 ± 0.02	0.89 ± 0.02
JNK3	0.92 ± 0.00	0.94 ± 0.01
Median 1	0.39 ± 0.02	0.39 ± 0.02
Median 2	0.36 ± 0.01	0.37 ± 0.03
Mestranol sim.	0.98 ± 0.00	0.98 ± 0.00
Osimertinib MPO	0.89 ± 0.01	0.89 ± 0.01
Perindopril MPO	0.74 ± 0.01	0.70 ± 0.03
QED	0.95 ± 0.00	0.95 ± 0.00
Ranolazine MPO	0.83 ± 0.00	0.81 ± 0.01
Scaff. hop	0.71 ± 0.10	0.68 ± 0.09
Sitagliptin MPO	0.58 ± 0.13	0.66 ± 0.05
Thiothixene redisc.	0.79 ± 0.06	0.83 ± 0.02
Troglitazone redisc.	0.72 ± 0.11	0.84 ± 0.02
Valsartan SMARTS	0.82 ± 0.19	0.90 ± 0.03
Zaleplon MPO	0.60 ± 0.01	0.60 ± 0.01
Sum	18.64 ± 0.32	18.88 ± 0.08

H.4 GOAL-DIRECTED SUBSTRUCTURE-CONSTRAINED DESIGN

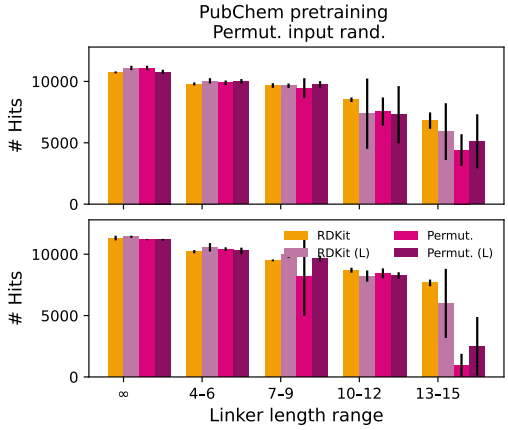
H.4.1 FRAGMENT LINKING



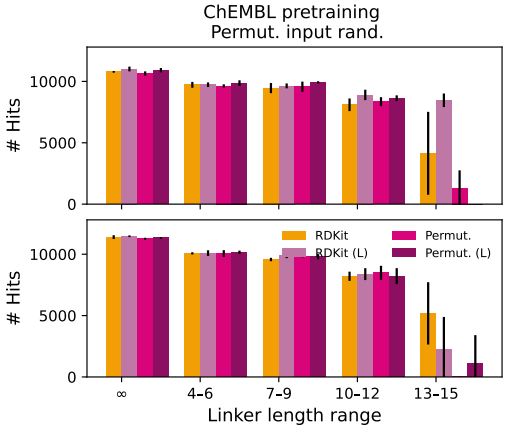
(a) PubChem pretraining, RDKit input randomization.



(b) ChEMBL pretraining, RDKit input randomization.



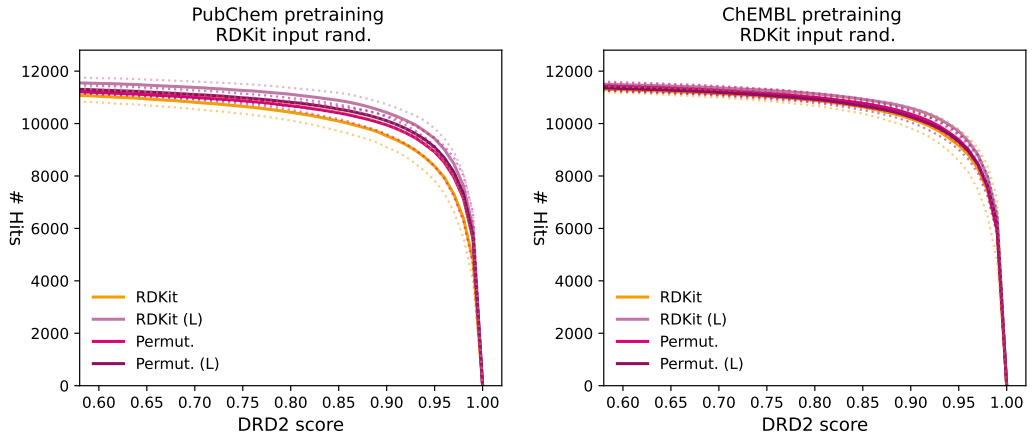
(c) PubChem pretraining, permutation input randomization.



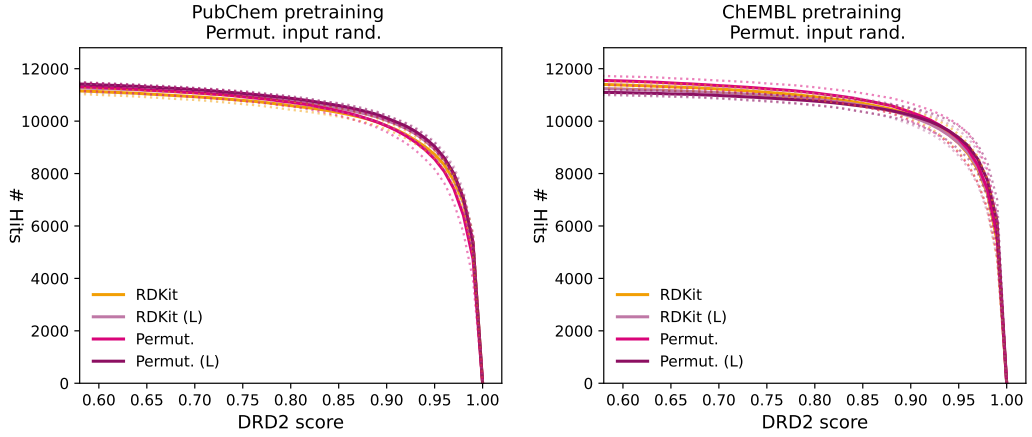
(d) ChEMBL pretraining, permutation input randomization.

Figure 10: Results for the goal-directed fragment linking experiments for OmniInvent and OmniInvent$_L$ pretrained on PubChem and ChEMBL with RDKit and permutation randomization. Inputs were randomized with either RDKit or permutation randomization. Labels indicate training-time randomization schemes, with L denoting the larger model variant.

H.4.2 SCAFFOLD DECORATION



(a) PubChem pretraining, RDKit input randomization. (b) ChEMBL pretraining, RDKit input randomization.



(c) PubChem pretraining, permutation input randomization. (d) ChEMBL pretraining, permutation input randomization.

Figure 11: Results for the goal-directed scaffold decoration experiment for OmniInvent and OmniInvent$_L$ pretrained on PubChem and ChEMBL with RDKit and permutation randomization. Inputs were randomized with either RDKit or permutation randomization. Labels indicate training-time randomization schemes, with L denoting the larger model variant.

I ABLATIONS

All ablations were trained using the ChEMBL-derived dataset with training-time RDKit randomization using the same hyperparameters as the 28-million-parameter OmniInvent model. We then evaluated the ablated models on direct sampling in the substructure-constrained design scenario using the same setup as in the main manuscript.

To investigate the importance of aligning masking patterns on SMILES to the underlying graph structure we conducted an experiment where training examples were generated by randomly masking out spans in the input SMILES sequence. To ensure that we probe graph alignment rather than the effect of span length distributions, we take the number of spans and the number of tokens in each span from the graph masking algorithm. The gaps between the masking spans are then drawn randomly.

To determine whether or not the mixing of de novo with substructure-constrained training examples has a detrimental impact on OmniInvent, we trained a model with $p_{\text{full}} = 0$, i.e., using only the graph masking strategy.