13 August 2026

Controllable molecular graph generation from natural-language chemical constraints

Li-Cheng Xu[1], Yu-Duo Qian[2,1], Fenglei Cao[1,3], Yuan Qi[1,2]

1. Shanghai Academy of Artificial Intelligence for Science
2. Artificial Intelligence Innovation and Incubation Institute, Fudan University
3. Golab Inc.

Abstract

Designing molecules under user-specified constraints is central to drug design, catalyst development and materials discovery. Deep generative models have advanced conditional molecule generation, but many controllable systems still rely on structured numerical or categorical inputs, limiting their ability to capture the qualitative, compositional and often ambiguous design intents used by chemists. Here, we report MolWeaver, a text-guided molecular graph generation framework that translates qualitative and quantitative natural-language prompts into chemically valid 2D molecular graphs. MolWeaver combines a frozen language-model backbone with explicit constraint grounding, graph-level control and scaffold-aware graph decoding. We evaluate MolWeaver on ZINC22-derived prompt-conditioned generation and downstream control tasks including ring-size specification, Lipinski rule-of-five-guided generation and phosphine ligand generation under specified steric demands. Across these settings, MolWeaver shows that natural language can serve as a practical interface for controllable molecular graph generation.

Controllable molecular graph generation from natural-language chemical constraints

Li-Cheng Xu,[1*] Yu-Duo Qian[1,2], Fenglei Cao,[1,3] Yuan Qi[1,2]

[1]Shanghai Academy of Artificial Intelligence for Science, Shanghai, 200232, China

[2]Artificial Intelligence Innovation and Incubation Institute, Fudan University, Shanghai, 201203, China

[3]Golab Inc., Shanghai, 200000, China

[*]Email: xulicheng@sais.org.cn

Abstract: Designing molecules under user-specified constraints is central to drug design, catalyst development and materials discovery. Deep generative models have advanced conditional molecule generation, but many controllable systems still rely on structured numerical or categorical inputs, limiting their ability to capture the qualitative, compositional and often ambiguous design intents used by chemists. Here, we report MolWeaver, a text-guided molecular graph generation framework that translates qualitative and quantitative natural-language prompts into chemically valid 2D molecular graphs. MolWeaver combines a frozen language-model backbone with explicit constraint grounding, graph-level control and scaffold-aware graph decoding. We evaluate MolWeaver on ZINC22-derived prompt-conditioned generation and downstream control tasks including ring-size specification, Lipinski rule-of-five-guided generation and phosphine ligand generation under specified steric demands. Across these settings, MolWeaver shows that natural language can serve as a practical interface for controllable molecular graph generation.

Introduction

Molecular design is increasingly driven by the need to satisfy explicit functional, structural and physicochemical requirements across drug discovery[1], catalyst development[2] and functional materials research[3,4]. In practical molecular design, chemists rarely seek molecules that are merely valid or novel. Instead, they often require candidates that satisfy multiple design intents simultaneously, spanning molecular composition, topology, physicochemical profiles and application-specific structural features[5]. These requirements may be expressed as precise numerical targets[6], but they are also frequently qualitative, approximate or context dependent[7]. Building generative models that can translate such mixed design language into chemically valid molecular structures remains an important challenge for AI-assisted molecular design[8].

Over the past decade, deep learning has become a central tool in chemical research[9–11], enabling advances in property prediction[12–14], reaction modeling[15–18], and molecular design[19–21]. A major consequence of this progress has been the rise of generative models for *de novo* molecular design[19]. Conditional generative models, in particular, have enabled molecules to be optimized or sampled under user-defined objectives[6,8], including drug-likeness[19], synthetic accessibility[19,22], binding-related scores[23], and other molecular properties[24,25]. These models have provided important demonstrations in drug-like molecule design[19], materials discovery[3,4,26] and catalyst optimization[2,24]. However, many controllable molecular generators still rely on structured inputs, such as scalar property values[6], categorical labels[19] or predefined reward functions[24]. Such interfaces are effective when the design objective can be

reduced to a small set of numerical targets, but they are less natural for the way chemists often describe molecules: as combinations of quantitative constraints, qualitative preferences and loosely specified structural intentions.

Natural-language offers a more flexible interface for molecular design[10,11]. A prompt can combine exact constraints, approximate ranges, qualitative descriptors and high-level design goals in a single instruction. Recent progress in large language models (LLMs) and multimodal generative models has shown that natural-language can act as a powerful conditioning signal for complex generation tasks, including text-to-image[27–29] and text-to-video generation[30–32]. In these domains, models can respond to both precise and ambiguous descriptions by learning representations that connect language with structured outputs. This suggests a similar opportunity for molecular generation: rather than requiring users to manually encode every objective as a numerical vector or reward function, a model could directly interpret natural-language chemical prompts and generate structures that reflect the intended properties.

Several recent studies have begun to explore language-guided molecular generation[33,34]. Sequence-to-sequence models translate textual descriptions into SMILES strings[35]; molecular language models use SMILES or SELFIES as the generation medium[36]; LLM-based systems use supervised fine-tuning or preference optimization to improve prompt following[34,37]; and emerging graph-diffusion methods[38] attempt to connect textual instructions with molecular graph generation. In parallel, molecule-text representation models have shown that contrastive alignment between molecular structures and textual descriptions can support text-based retrieval and molecule editing[39]. These studies demonstrate the promise of natural-language molecular design, but important limitations remain. String-based methods inherit the indirectness of SMILES generation: global graph properties such as ring count, heteroatom composition or local topology must be represented through a linear sequence. LLM-based generators can provide a convenient instruction interface, but the mapping from prompt semantics to chemical structure is often implicit. Graph-based diffusion methods better represent molecular topology, yet many of them are designed primarily for numerical or categorical property conditioning rather than mixed qualitative and quantitative language. A key unmet need is therefore a model that combines the semantic flexibility of language-based conditioning with explicit, chemically meaningful control over molecular graph generation.

Here we report MolWeaver, a hybrid deep learning architecture for controllable molecular graph generation from qualitative and quantitative natural-language prompts. MolWeaver combines a frozen language-model text encoder with a condition adaptation and constraint parsing module, a constraint compiler, a scaffold-aware prior, and a discrete 2D molecular graph denoising decoder. Instead of treating a prompt as a single undifferentiated text embedding, MolWeaver separates general semantic tokens from property constraint tokens and injects them into the graph decoder through distinct conditioning channels. It parses natural-language prompts into explicit property slots and compiles these slots into graph-level structural budgets, providing a general mechanism for translating qualitative and quantitative design intents into controllable graph-generation signals. The model further incorporates text-graph alignment to couple textual intent with molecular graph representations, and introduces scaffold priors to guide global topology before atom- and bond-level decoding. Together, these designs allow MolWeaver to convert mixed natural-language instructions into controllable graph-generation signals.

We pretrain MolWeaver on more than 2.2 million ZINC22-derived[40] molecule-prompt pairs generated from 881,038 neutral small molecules. As a proof of concept, the pretraining corpus contains prompts with mixed quantitative and qualitative conditions, instantiating this framework with five numerical constraints over logP,

heavy atom count, halogen count, ring count and heteroatom count, together with five fuzzy natural-language descriptions of lipophilicity, flexibility, branching, ring density and heterocycle profile. Each prompt combines multiple conditions, enabling the model to learn multi-constraint molecular generation rather than single-property optimization alone. We then evaluate MolWeaver in three downstream settings chosen to probe both transferability and practical utility. Ring-size and ring-count control tests whether simple fine-tuning can extend the model to constraints that are related to, but not explicitly included among, the ten proof-of-concept pretraining conditions, whereas Lipinski rule-of-five[41] generation and phosphine ligand generation with specified local steric demands demonstrate application-oriented control in drug-like molecule design and catalyst ligand design, respectively. These studies show that natural-language prompts can serve as a practical and expressive interface for controllable molecular graph generation, bridging human chemical intent and structure-level molecular design.

Results

Architecture of MolWeaver

To realize this natural-language interface for molecular design, we developed MolWeaver as a hybrid text-to-graph architecture that couples prompt understanding with explicit molecular constraint control (Fig. 1). The model takes a natural-language prompt as input and generates a discrete 2D molecular graph, including atom types, bond types, node presence, chirality and bond stereochemistry. Rather than treating the prompt as a single undifferentiated conditioning vector, MolWeaver preserves broad semantic information while routing property-specific constraint information through a dedicated constraint-grounding pathway before graph generation.

MolWeaver first encodes each prompt with Qwen3.5-0.8B text encoder[42] and projects the resulting token embeddings into the hidden space of the molecular generator (Fig. 1a). The projected prompt tokens are then processed by two parallel modules. A semantic condition adapter compresses the prompt into condition tokens that preserve the overall design intent. In parallel, a constraint parser uses property-slot attention to extract property-specific slot tokens from the same prompt representation (Fig. 1b). Slot prediction heads estimate whether each property is mentioned, its target value or category, and its constraint hardness. These outputs are used to form hard and soft constraint tokens, allowing the model to separate quantitatively specified constraints from softer qualitative descriptions during graph generation.

The parsed constraints are then passed to a graph constraint compiler (Fig. 1b), which bridges property-level prompt interpretation and graph-level molecular construction. This module integrates the property slot tokens, hard and soft constraint tokens, and predicted property values to produce compiled constraint tokens and graph-control budgets. These budgets translate language-derived constraints into graph-level quantities that can directly guide molecular generation, such as molecular size, atom-composition-related constraints and ring-topology-related constraints. This design makes the constraint pathway chemically interpretable while retaining the flexibility of natural-language prompting.

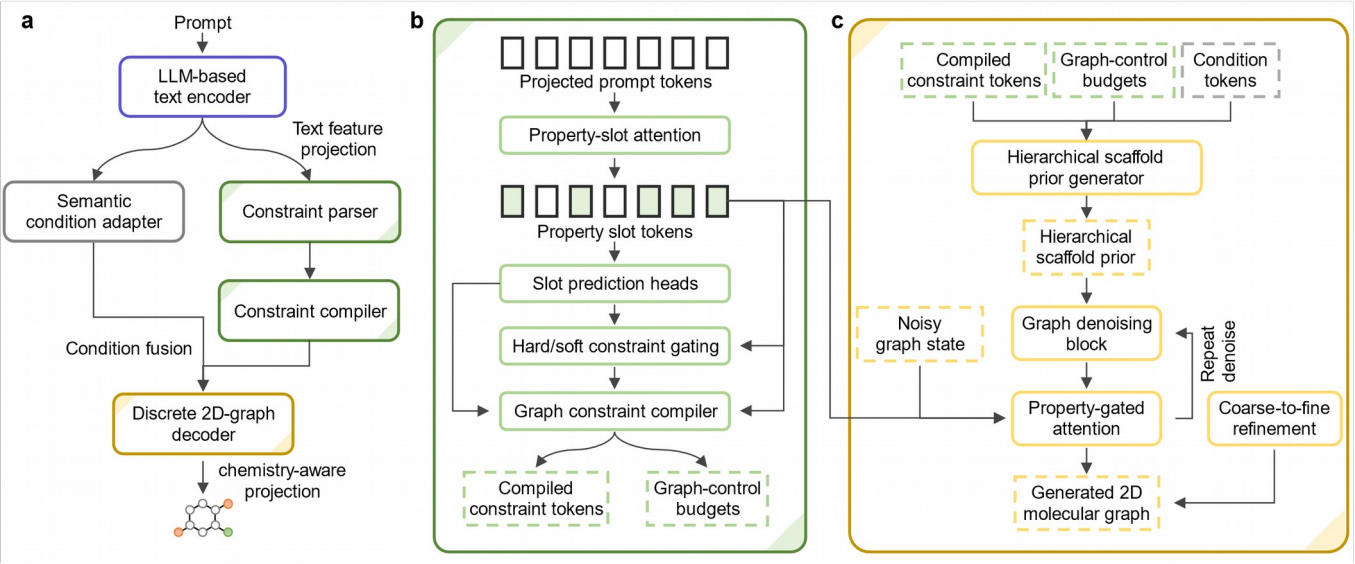

Fig. 1 | MolWeaver architecture for natural-language-conditioned molecular graph generation. a, Overview of MolWeaver. A natural-language prompt is encoded by an LLM-based text encoder and projected into the generator hidden space, where semantic condition tokens and explicit constraint tokens are extracted through separate pathways. Parsed constraints are compiled into graph-control signals and used to condition discrete 2D molecular graph generation. **b,** Details of the constraint-grounding module, showing how projected prompt tokens are parsed into property slots and transformed into hard and soft constraint tokens, compiled constraint tokens and graph-control budgets. **c,** Details of the graph-decoding module, showing how semantic condition tokens, compiled constraints and graph-control budgets are combined with a hierarchical scaffold prior to guide discrete 2D graph denoising and molecular graph prediction.

The molecular graph is generated by a discrete 2D graph denoising decoder (Fig. 1c). Before denoising, a hierarchical scaffold prior generator uses the semantic condition tokens, compiled constraint tokens and graph-control budgets to predict scaffold-level priors over node presence, atom identity and bond structure. These priors provide a coarse structural blueprint that biases the subsequent denoising process toward graphs with compatible size, composition and topology. The denoising decoder then iteratively refines a noisy graph state. At each step, it cross-attends to the condition and compiled constraint tokens, while a property-gated attention pathway separately injects the property slot tokens into the graph states. A coarse-to-fine refinement module updates node representations using predicted atom and edge distributions before producing final graph logits.

In addition to graph reconstruction and property-control losses, we include a text-graph alignment loss during training. Similar in spirit to text-image contrastive alignment in text-to-image generation[43], this objective encourages natural-language prompts and molecular graphs to share a compatible representation space. This design reflects the many-to-many nature of language-guided molecular design: a single prompt can correspond to multiple chemically valid molecular graphs, and a single molecular graph can be described by multiple prompts with different emphases. We therefore use multi-positive contrastive alignment to bring prompt and graph embeddings with compatible property constraints closer together, while separating incompatible prompt-graph pairs. This loss provides an additional training signal that links natural-language intent to generated graph structure, complementing the explicit constraint parsing and graph-control pathways.

Together, these designs allow MolWeaver to connect natural-language chemical intent to discrete molecular graph construction. The semantic pathway captures flexible prompt meaning, the constraint pathway converts explicit and fuzzy property descriptions into graph-control signals, and the scaffold and denoising modules use these signals to construct chemically valid molecular graphs. This architecture provides the basis for the multi-constraint pretraining and downstream control experiments described below.

Construction of molecule-prompt corpora

To train MolWeaver to follow natural-language chemical intent rather than isolated numerical labels, we constructed molecule-prompt corpora that pair molecular graphs with textual descriptions of desired properties and structural features. The corpora were designed to support two goals: first, to teach the model general multi-condition molecular generation from mixed quantitative and qualitative prompts; and second, to test whether this capability can be adapted to new constraints and practical design scenarios through fine-tuning.

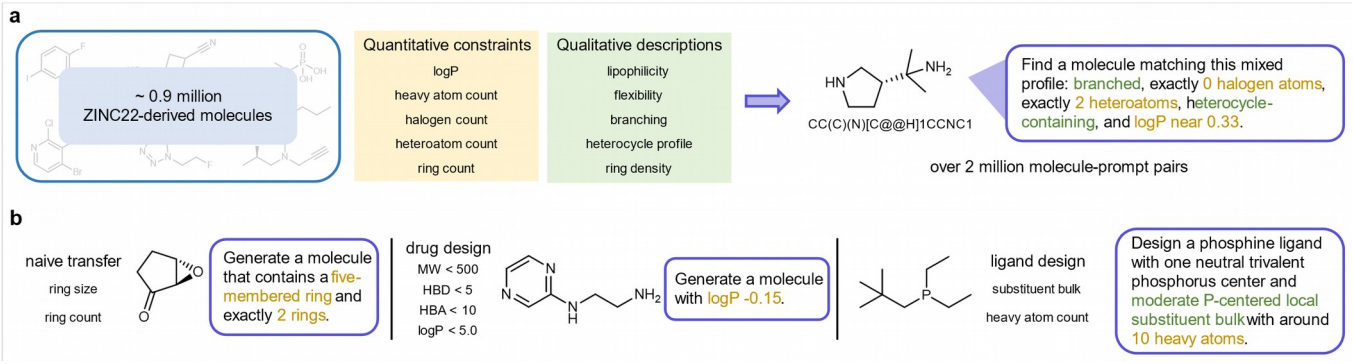

Fig. 2 | Construction of molecule-prompt corpora for pretraining and downstream fine-tuning. a, Construction of the pretraining corpus. Approximately 0.9 million ZINC22-derived small molecules were paired with natural-language prompts generated from five quantitative molecular properties and five qualitative molecular descriptions, yielding more than 2 million molecule-prompt pairs for pretraining. **b,** Construction of the fine-tuning corpora. Three downstream corpora were prepared to evaluate transfer to ring-size control, Lipinski rule-of-five-guided generation and phosphine ligand generation under specified steric demands.

For pretraining, we assembled a ZINC22-derived library[40] of nearly 0.9 million neutral small molecules with 4-20 heavy atoms and element types restricted to C, N, O, F, P, S, Cl, Br and I (Fig. 2a). From these molecules, we generated natural-language prompts that combine multiple design conditions. As a proof of concept, the pretraining prompts covered five quantitative constraints, including logP and molecular composition or topology, together with five qualitative descriptions related to lipophilicity, flexibility, branching, ring density and heterocycle profile. After balancing over prompt patterns and molecular scaffolds, the final pretraining corpus contained over 2 million molecule-prompt pairs. This design exposes MolWeaver to prompts that resemble practical molecular design requests, where several quantitative targets and qualitative preferences are often specified together.

We further prepared three fine-tuning corpora to evaluate transferability and application-oriented control. A ring-control corpus, comprising 2,000 molecules and 12,080 molecule-prompt pairs, tested adaptation to ring-size specification (left panel of Fig. 2b), a structural constraint not included in the proof-of-concept pretraining conditions, and its combination with pretrained ring-count control. The Lipinski rule-of-five (RO5) corpus

comprised 2,000 RO5-compliant molecules and 8,000 molecule-prompt pairs, and evaluated drug-like molecule generation by requiring molecules to satisfy RO5 criteria while following one explicit quantitative RO5-related constraint, such as molecular weight (MW), hydrogen-bond donor (HBD) count, hydrogen-bond acceptor (HBA) count, or logP (middle panel of Fig. 2b). Finally, the phosphine ligand corpus was constructed from the Kraken phosphine ligand dataset[44] by extracting substituent fragments from single-phosphine ligands and recombining them to generate an expanded trialkyl phosphine library. From this library, we selected 7,712 neutral trivalent trialkyl phosphines with no more than 20 heavy atoms, yielding 15,424 molecule-prompt pairs. This corpus represented catalyst ligand design, with prompts specifying a neutral trivalent phosphorus center, local steric bulk around phosphorus and approximate molecular size (right panel of Fig. 2b). Together, these datasets allow us to evaluate MolWeaver from general language-conditioned graph generation to focused downstream molecular design tasks.

Pretraining enables multi-condition molecular control

We first evaluated whether MolWeaver could follow mixed natural-language constraints after pretraining alone. On a held-out test set of 1,000 prompts, we compared MolWeaver with three representative LLM baselines, GPT-5[45], Qwen3.7-plus[42] and DeepSeek-V4-flash-preview[46], which were prompted to directly generate SMILES strings from the same instructions (Fig. 3a). Despite this comparison against much larger general-purpose LLMs, MolWeaver contains only approximately 0.8 billion parameters and can be deployed on consumer-grade GPUs.

Across the five quantitative and five qualitative pretraining conditions, MolWeaver showed consistently strong prompt adherence (Fig. 3a, left). For quantitative constraints, MolWeaver achieved 100.0% adherence for heavy atom count, 98.1% for heteroatom count, 93.4% for halogen count, 91.7% for ring count and 74.0% for logP. For qualitative descriptions, it achieved 88.2% adherence for lipophilicity, 92.0% for flexibility, 87.3% for branching, 94.8% for ring density and 92.1% for heterocycle profile. The LLM baselines were less reliable, especially for quantitative control. Qwen and DeepSeek performed poorly on several numerical constraints, including logP, heavy atom count and heteroatom count, whereas GPT-5 was the strongest LLM baseline but still trailed MolWeaver on most conditions. LogP was the most difficult condition for all methods, consistent with the fact that this property is not directly readable from simple graph counts or local topology. MolWeaver addresses this difficulty by including a logP prediction pathway during training, which helps convert this non-local physicochemical property into a graph-level control signal. To further probe whether LLM outputs reflected generative exploration or repeated retrieval-like behavior, we also repeated generation 100 times for a single six-condition prompt. In this setting, GPT-5 showed substantially lower uniqueness than MolWeaver, suggesting a narrower output distribution despite repeated sampling (Supplementary Fig. 5).

We next examined how prompt adherence changed as the number of simultaneous constraints increased (Fig. 3a, middle). MolWeaver achieved an overall strict prompt adherence rate of 67.5% across the 1,000 test prompts. As expected, performance decreased with constraint complexity, from 76.0% for two-condition prompts to 72.0%, 65.0%, 63.0% and 61.5% for three-, four-, five- and six-condition prompts, respectively. GPT-5 again outperformed Qwen and DeepSeek among the LLM baselines, with an overall adherence rate of 56.5%, but remained below MolWeaver, particularly for six-condition prompts (44.5% for GPT-5 versus 61.5% for MolWeaver). Qwen and DeepSeek achieved much lower overall adherence rates of 16.0% and 12.7%, respectively, indicating that direct SMILES generation by general-purpose LLMs struggles to maintain multiple simultaneous

molecular constraints. Representative prompt-adherent and non-adherent generations from the pretrained MolWeaver model are shown in Supplementary Fig. 1.

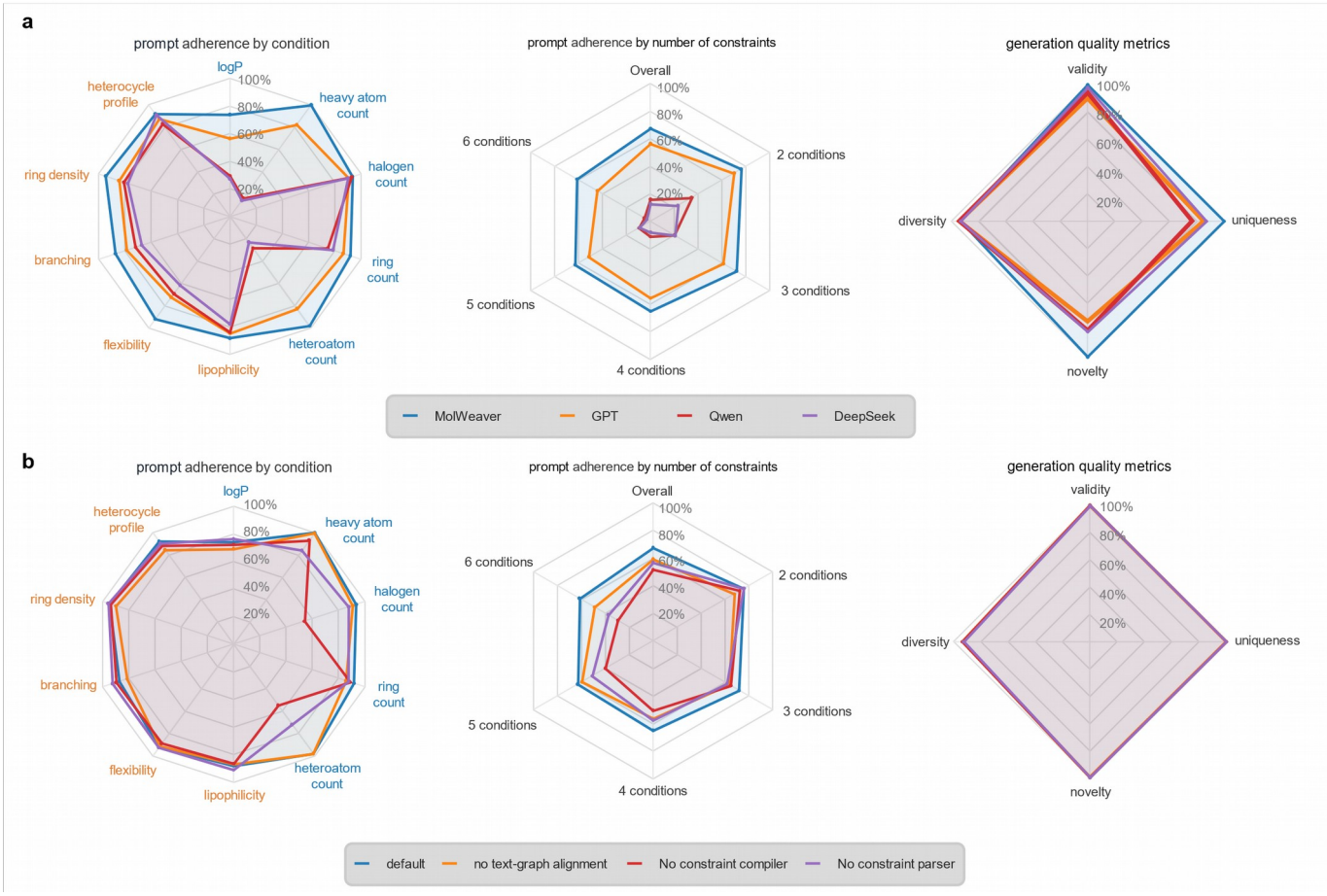

Fig. 3 | Pretraining enables multi-condition molecular control and identifies the contribution of constraint-aware modules. a, Comparison of MolWeaver with three representative LLM baselines, GPT-5, Qwen3.7-plus and DeepSeek-V4-flash-preview, on the held-out pretraining test set. From left to right, radar plots show prompt adherence for the five quantitative and five qualitative pretraining conditions, strict prompt adherence stratified by the number of simultaneous prompt constraints, and molecular generation quality measured by validity, uniqueness, novelty and internal diversity. **b,** Ablation analysis on the same test set, with the same three readouts as in a. Curves compare the full MolWeaver model with variants lacking text-graph alignment, the constraint compiler or the constraint parser. Novelty is computed relative to the MolWeaver pretraining corpus.

MolWeaver also maintained strong molecular generation quality (Fig. 3a, right). It achieved 100.0% RDKit validity, 100.0% uniqueness, 99.4% novelty relative to the MolWeaver pretraining corpus and an internal diversity of 93.0%. By comparison, the LLM baselines showed lower uniqueness, particularly GPT-5 and Qwen, indicating more repeated structures in their generated sets. Novelty for LLMs should be interpreted only as novelty relative to the MolWeaver pretraining corpus, rather than as model-specific training-set novelty, because the training data of the proprietary or external LLMs are not known.

Ablation experiments further supported the architectural design of MolWeaver (Fig. 3b). Removing text-graph alignment reduced overall prompt adherence from 67.5% to 59.3%, removing the constraint parser reduced it to 56.8%, and removing the constraint compiler caused the largest drop, to 51.5%. The same trend was most pronounced for higher-order prompts: for six-condition prompts, adherence decreased from 61.5% in the full model to 49.0% without text-graph alignment, 37.5% without the constraint parser and 29.5% without the constraint compiler. These results indicate that all three components contribute to translating natural-language instructions into graph-level control, with the constraint compiler playing the largest role in multi-condition satisfaction. In contrast, the ablations had little effect on validity, uniqueness, novelty or diversity, which remained close to the full model. Thus, these modules primarily improve controllability rather than merely increasing chemical validity or output diversity, supporting the overall design of MolWeaver as a language-guided, constraint-aware molecular graph generator.

Fine-tuning extends molecular control

After establishing that pretraining enables MolWeaver to follow mixed quantitative and qualitative prompts, we next asked whether this capability could be transferred to new structural constraints and application-oriented molecular design tasks through lightweight fine-tuning. We considered three downstream settings: ring-size control as a transferability test, Lipinski rule-of-five generation as a drug-like molecule design task, and phosphine ligand generation with specified local steric demand as a catalyst-ligand design task.

The ring-control task was designed to test whether MolWeaver could adapt to a structural constraint not explicitly included among the proof-of-concept pretraining conditions. The corpus contained 12,080 prompts from 2,000 ZINC22-derived molecules, covering ring-size prompts, ring-count prompts and their combinations. After fine-tuning, MolWeaver generated RDKit-valid molecules for 100.0% of held-out test prompts. It satisfied 96.2% of ring-size-only prompts and 79.6% of prompts requiring both a specified ring size and ring count; ring-count-only prompts were satisfied at 99.7%. These results indicate that the pretrained model can acquire a new graph-topological control mode through fine-tuning, while combined structural constraints remain more challenging than single-condition prompts.

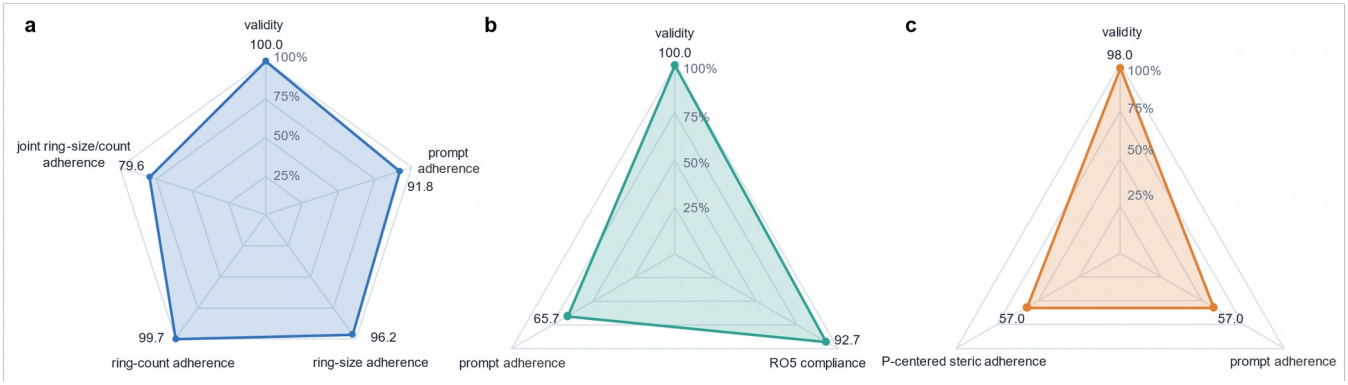

Fig. 4 | Fine-tuning extends molecular control to downstream design tasks. a, Performance after fine-tuning for ring-size control, a transferability task evaluated by validity, prompt adherence, ring-size adherence, ring-count adherence and joint ring-size/count adherence. **b,** Performance after fine-tuning for Lipinski rule-of-five-guided generation, evaluated by validity, prompt adherence and RO5 compliance. **c,** Performance after fine-tuning for

phosphine ligand generation under specified local steric demands, evaluated by validity, prompt adherence and P-centered steric adherence.

We then evaluated drug-like molecule generation using a Lipinski rule-of-five corpus. This dataset contained 8,000 prompts from 2,000 RO5-satisfying ZINC22 molecules, where each prompt required the generated molecule to satisfy the RO5 criteria while also following one explicit quantitative condition, including molecular weight, hydrogen-bond acceptor count, hydrogen-bond donor count or logP. MolWeaver generated RDKit-valid molecules for 100% of held-out test prompts, and 92.7% of generated molecules satisfied the RO5 criteria. The specific quantitative prompt requirement was satisfied for 65.7% of prompts. This separation between high RO5 compliance and lower exact quantitative matching suggests that MolWeaver can reliably stay within a drug-like chemical regime, while precise scalar control over individual medicinal-chemistry properties remains a harder target.

Finally, we evaluated phosphine ligand generation for catalyst design. The phosphine corpus contained 15,424 prompts from 7,712 expanded trialkyl phosphines, with prompts specifying a single neutral trivalent phosphorus center, local steric bulk around phosphorus and approximate molecular size. MolWeaver generated RDKit-valid structures for 98.0% of held-out prompts. The molecular-size requirement was satisfied in all valid connected generations, whereas the P-centered local steric class was the main limiting constraint. Consequently, 57.0% of all test generations satisfied the complete ligand-design prompt. This task is substantially more specialized than the ZINC22 pretraining setting and is built from a chemically distinct ligand dataset rather than ZINC22-like drug-like molecules. Nevertheless, MolWeaver retained high chemical validity and successfully transferred its language-guided graph-generation capability to phosphine ligand design, learning to bias generated ligands toward specified local steric environments. Representative prompt-adherent and non-adherent generations from the three fine-tuned models are shown in Supplementary Figs. 2–4.

Together, these downstream experiments show that MolWeaver is not limited to the ten proof-of-concept pretraining conditions. The same text-to-graph framework can be adapted to new structural constraints and to practical design settings in drug-like molecule generation and catalyst ligand design. This supports the central premise of MolWeaver: natural language can serve as a flexible interface for molecular graph generation, while explicit constraint parsing, graph-level control and chemistry-aware decoding provide the structure needed to turn human design intent into valid molecular candidates.

Discussion

We developed MolWeaver to make molecular graph generation controllable through natural-language prompts that may contain both quantitative constraints and qualitative design intent. The central design choice is to avoid treating a prompt as a single opaque conditioning vector. Instead, MolWeaver separates general semantic information from property-specific constraint information, parses natural-language constraints into explicit property slots, compiles them into graph-control budgets and constraint tokens, and injects these signals into a discrete 2D graph denoising decoder. This architecture was designed to combine the semantic flexibility of language models with chemically interpretable graph-level control.

On the pretraining task, MolWeaver generated chemically valid, diverse and mostly novel molecules while maintaining substantial multi-condition prompt control. The model achieved 100.0% RDKit validity, 67.5% strict

prompt success, 99.4% novelty and 93.0% internal diversity on the held-out pretraining test set. Importantly, performance remained meaningful as prompts increased from two to six simultaneous conditions, showing that the model learned multi-constraint generation rather than isolated single-property matching. Compared with pure LLM baselines that directly generate SMILES from the same prompts, MolWeaver is expected to provide a stronger route to chemical validity and constraint satisfaction because it generates molecular graphs through chemistry-aware denoising and explicit graph-control signals rather than relying only on implicit sequence-level instruction following. Another practical advantage is model size: MolWeaver uses an approximately 0.8B-parameter language backbone and requires less than 8 GB of GPU memory for inference, enabling deployment on consumer-grade GPUs, whereas the strongest general-purpose LLM baselines typically require much larger proprietary or server-scale models. The final quantitative comparison with GPT-5, Qwen-3.7-plus and DeepSeek-V4-flash-preview will further clarify this advantage.

The ablation experiments support the importance of the main architectural components. Removing text-graph alignment tests whether prompt and molecular graph embeddings benefit from an explicit many-to-many alignment objective. Removing the constraint parser tests whether property-slot extraction is needed beyond generic text conditioning, and removing the constraint compiler tests whether parsed prompt constraints should be translated into graph-level budgets before generation. All three ablations reduced prompt adherence, indicating that each component contributes to controllable generation. The largest degradation was observed when the constraint compiler was removed, showing that graph-control budgets provide information beyond slot-level property tokens alone and are particularly important for satisfying multiple simultaneous constraints. By contrast, validity, uniqueness, novelty and diversity remained largely unchanged across ablations, suggesting that these modules primarily improve language-to-graph controllability rather than general chemical validity or output diversity.

The fine-tuning experiments show that MolWeaver can transfer beyond the proof-of-concept pretraining conditions. Ring-size fine-tuning demonstrated adaptation to a new topological control task related to, but not explicitly included in, pretraining. Lipinski rule-of-five fine-tuning showed that the model can generate drug-like molecules under medicinal-chemistry constraints, maintaining high RO5 compliance while following explicit quantitative prompts. The phosphine ligand task further extended MolWeaver to a chemically distinct ligand dataset derived from Kraken phosphines, where the model learned to generate neutral trivalent trialkyl phosphines with specified local steric demand and approximate molecular size. These results suggest that the pretrained language-conditioned graph generator can be redirected toward new chemical design regimes with task-specific fine-tuning.

Overall, MolWeaver illustrates a path toward molecular design systems in which chemists specify intent in natural language while the model translates that intent into chemically meaningful graph-generation controls. Its value is not simply that it accepts text input, but that it connects language, property constraints and molecular graph construction through an interpretable architecture. This makes it a step toward more flexible molecular generation workflows, where qualitative preferences, quantitative targets and domain-specific structural requirements can be combined in a single prompt and optimized within a chemically valid graph space. Future work should extend the range of supported properties, strengthen precise continuous-property control and connect generated candidates to downstream experimental or simulation-based validation.

Methods

PyTorch (v.2.4.0) was used for model implementation, training and inference. NumPy (v.1.26.4) and Pandas (v.2.3.3) were used for numerical operations, data processing and result analysis. RDKit (v.2025.09.3) was used for molecular parsing, canonical SMILES generation, descriptor calculation, validity checking, molecular fingerprinting and structure visualization.

Model architecture

The overall architecture of MolWeaver is shown in Fig. 1. Given a textual molecule description, the input is first encoded by the frozen Qwen3.5-0.8B backbone into contextual embeddings and then projected through a shared linear layer into two parallel branches. In the semantic branch, the projected token embeddings are compressed into semantic condition tokens that provide coarse-grained textual guidance for subsequent graph generation. In the constraint branch, learnable property slots extract fine-grained chemical property constraints from the text. The outputs of the constraint branch are further compiled into explicit graph-control budgets (atom counts, ring counts, etc.) that directly constrain the molecular topological space. On the generation side, a hierarchical scaffold prior generator first produces a coarse molecular scaffold from the conditions and budgets, and a discrete 2D graph denoising decoder then uses the scaffold as a prior and the semantic and constraint embeddings as conditions to generate the final molecular graph through coarse-to-fine iterative refinement. In addition, a contrastive learning module assists the training of the text and graph encoders by aligning the representations of text descriptions and generated molecules in embedding space.

Text encoding

We use the frozen Qwen3.5-0.8B as the text encoder. The input text is encoded into token-level embeddings and a sentence-level pooled embedding, which are then projected into a 512-dimensional hidden space through a shared linear layer. The projected token embeddings are further compressed by a Transformer module into a fixed set of 16 semantic condition tokens, reducing the variable-length textual information into a compact representation and lowering the computational cost of subsequent cross-attention in the decoder.

Constraint parsing and compilation

Reliably extracting numerical chemical constraints directly from text (e.g., "logP around 2.5", "containing one six-membered aromatic ring") is essential for generating chemically valid molecules. To this end, we design a set of 24 learnable property slots, each corresponding to one chemical property (heavy atom count, logP, ring count, etc.), which extract relevant information from the text embeddings through multi-layer cross-attention. Each slot outputs four predictions: the property value, the mention probability (whether the text mentions the property), the constraint hardness (hard or soft constraint), and a classification distribution over discretized value buckets, the latter improving the precision of numerical regression. The pooled slot embeddings yield a global property embedding.

Based on the differentiated control logic that quantitative hard constraints must be strictly satisfied whereas qualitative soft constraints provide design preferences, we split the slot embeddings $Q \in R^{N \times d}$ into two constraint embeddings according to the hardness probability (Eqs. 1, 2):

$$C_{\text{hard}} = Q \odot p^{(\text{mention})} \odot p^{(\text{hard})} \tag{1}$$

$$C_{\text{soft}} = Q \odot \left(1 - p^{(\text{hard})}\right) \tag{2}$$

where $C_{\text{hard}}, C_{\text{soft}} \in R^{N \times d}$ denote the hard and soft constraint embeddings, respectively, $\odot$ denotes element-wise multiplication, and $p^{(\text{mention})}$ and $p^{(\text{hard})}$ are the mention and hardness probabilities defined above. The hard-constraint branch is further refined by a cross-attention compiler into four graph-control budgets—a node budget, an atom budget, a heteroatom budget and a ring budget—that directly constrain the topological space of the generated molecules; the soft-constraint branch is passed to the graph decoder as conditioning embeddings, providing non-binding property preferences.

Hierarchical scaffold generation and graph decoding

To guide molecular graph construction with global structural information, MolWeaver first predicts a scaffold-level prior before discrete graph denoising. The hierarchical scaffold prior generator consumes semantic condition tokens, soft constraint tokens, compiled constraint tokens, the global prompt embedding and graph-control budgets derived from the constraint compiler. The node, atom-composition, heteroatom and ring budgets are normalized and projected into the decoder hidden space, then added to learnable node queries together with the global prompt embedding. These node states cross-attend to the concatenated condition and compiled constraint tokens and are refined by transformer layers to produce scaffold-level node-mask, atom-type and bond-type logits, as well as scaffold node states.

The graph decoder then performs conditional denoising over a fixed-size discrete molecular graph. At each denoising step, node states are initialized from learnable node queries, the global prompt embedding, the current noisy atom, bond, chirality, node-presence and timestep embeddings, and the scaffold node-state prior. The node states attend to condition and compiled constraint tokens for global prompt conditioning, while a property-specific cross-attention stream injects property slot tokens through a learned gate. The resulting node states are further refined by coarse-to-fine graph refinement blocks: coarse atom and bond distributions are predicted, transformed into atom and edge contexts, aggregated through predicted bond probabilities and used to update node states. Final atom, bond, node-presence, chirality and bond-stereochemistry logits are then produced and combined with the scaffold priors. During sampling, chemistry-aware projection steps enforce valence-compatible bond choices, stereochemical constraints, connectivity-related constraints and task-specific topology rules where applicable.

Text–graph contrastive alignment

To constrain the text encoder and molecular representations into a shared representation space, we introduce a multi-positive contrastive alignment loss so that conditional generation can be formulated as matching within this space rather than mapping across separate spaces. Unlike standard contrastive learning, which treats only exact matching pairs as positives, our method builds positive pairs within the same batch based on chemical property similarity: two molecules are considered a positive pair if they have the same set of mentioned properties and all shared property values match within tolerance. This design substantially increases the number of effective positive pairs within a batch.

Let κ be the temperature parameter and $s_{ij} = \kappa \cdot z_t^{(i)} \cdot z_g^{(j)}$ be the scaled cosine similarity between text i and graph j. The contrastive loss is defined as (Eq. 3):

$$L_{\text{align}} = \frac{-1}{B} \sum_{i=1}^{B} \frac{1}{|P_i|} \sum_{j \in P_i} \log \frac{\exp(s_{ij})}{\sum_{k=1}^{B} w_{ik} \exp(s_{ik})} \tag{3}$$

where P_i is the set of positive graph indices for text i, and w_{ik} is the hard-negative weight: $w_{ik} > 1$ if text i and molecule k only partially match in properties, otherwise $w_{ik} = 1$.

Pretraining data and prompt-corpus construction

Data collection and filtering

The pretraining dataset was derived from ZINC22, a large collection of purchasable compounds. We selected molecules with 4–20 heavy atoms from curated ZINC22 SMILES pools and retained only neutral molecules containing allowed elements (C, N, O, F, P, S, Cl, Br and I). For molecules with 11–20 heavy atoms, we additionally restricted logP to the range −3 to 3 and capped the number of selected molecules from each ZINC22 tranche at 2,000 to improve structural diversity. After filtering and deduplication, the merged dataset contained 881,038 neutral small molecules.

Prompt-corpus generation

To build the text–molecule paired data required for pretraining, we generated natural-language prompts from descriptor-derived molecular properties. As a proof-of-concept prompt space, we used five quantitative properties, including logP, heavy atom count, halogen count, ring count and heteroatom count, and five qualitative descriptions, including lipophilicity, flexibility, branching, ring density and heterocycle profile. Each prompt combined 2–6 properties into a concise natural-language instruction, allowing the model to learn multi-condition molecular generation rather than single-property conditioning.

For each molecule, up to four prompt variants were generated by sampling different property subsets and phrasings while preserving descriptor consistency. The number of conditions per prompt was approximately balanced from two to six, and prompt generation was made deterministic from the canonical SMILES and prompt-variant index to ensure reproducibility.

Corpus balancing and splitting

To mitigate distribution bias in the prompt corpus, we apply a balancing procedure that limits the number of molecules per unique prompt template (at most 32) and per Bemis–Murcko scaffold (at most 8). This ensures that the model is exposed to diverse chemical structures for each prompt type and prevents overfitting to frequent scaffolds.

The balanced corpus is split into a training set (about 2.2 million molecule-prompt pairs), a validation set (1,000 pairs) and a test set (1,000 pairs) by deterministic stratified sampling over the number of prompt conditions with a fixed seed. For the validation and test sets, each molecule is associated with all of its prompt variants to allow comprehensive evaluation.

Downstream fine-tuning

We evaluate MolWeaver on three text-guided molecule generation tasks that require controlling specific structural properties. All tasks share the same fine-tuning pipeline: load the model from the pretrained checkpoint, fine-tune on the task-specific molecule–prompt corpus, and enable property guidance at inference. The corpus of each task is split into training, validation and test sets in an 8:1:1 ratio by a deterministic hash of the canonical SMILES.

1. Ring-size control. From the pretraining corpus, we select 2,000 molecules containing specified ring systems (four-, five- or six-membered rings), each associated with a prompt identifying its ring type (four-, five- or six-membered) and ring count. The model is fine-tuned on the 12,080 molecule-prompt pairs generated from these molecules.

2. Lipinski rule-of-five generation. From the pretraining corpus, we select 2,000 molecules that simultaneously satisfy the four Lipinski rules (molecular weight < 500 Da, logP < 5.0, hydrogen-bond donors < 5, hydrogen-bond acceptors < 10). For each molecule, four single-property quantitative prompts are generated for molecular weight, logP, hydrogen-bond donor count and hydrogen-bond acceptor count, e.g., "Generate a molecule with molecular weight 380.50 Da", with the four prompts drawn from three phrasings per property. The model is fine-tuned on 8,000 prompts.

3. Phosphine ligand generation. Starting from the Kraken phosphine ligand dataset, we first extracted 931 single neutral trivalent phosphines and decomposed them into 446 unique P-bound substituents. These substituents were recombined to construct an expanded phosphine library, yielding 21,492 candidate phosphines. We then selected 7,712 trialkyl phosphines with at most 20 heavy atoms and allowed elements, each containing a single neutral trivalent phosphorus center. For each molecule, two prompts were generated to specify the phosphorus-center requirement, the P-centered local steric bulk class (low, moderate or high) and the approximate molecular size, resulting in a corpus of 15,424 prompts.

Evaluation metrics

We use the following metrics to evaluate the generated molecules: chemical validity (the proportion of molecules passing RDKit valence, aromaticity and connectivity checks), uniqueness (the proportion of unique molecules after canonical-SMILES deduplication among valid generated molecules), novelty (the proportion of unique valid molecules not present in the training set) and internal diversity (the mean pairwise Tanimoto dissimilarity of the generated molecule set). The prompt adherence rate is defined as the proportion of generated molecules satisfying all constraints specified in the prompt, with determination rules and tolerances given in Supplementary Table 3.

Data availability

The pretraining and fine-tuning molecule-prompt corpora used in this study, together with the corresponding trained model weights, are available through https://www.modelscope.cn/models/XuLiCheng2025/MolWeaver. The released data include the ZINC22-derived pretraining corpus and the three downstream fine-tuning corpora for ring-size control, Lipinski rule-of-five-guided generation and phosphine ligand generation under specified steric demands.

Code availability

The source code for data preprocessing, model architecture definition, model training and inference is available at https://github.com/licheng-xu-echo/MolWeaver.

Acknowledgements

Generous support by the National Natural Science Foundation of China (82394432 and 92249302, Y.Q.), the Shanghai Municipal Science and Technology Major Project (Grant No. 2023SHZDZX02, Y.Q.), the Oriental Talent Youth Project (L.-C.X.), and the Shanghai AI for Science Hundred Teams Hundred Projects (F.C.). The AI model training was conducted using the Inspire platform at SAIS.

Author contributions

L.-C.X. proposed the study. L.-C.X. designed the model. L.-C.X. and Y.-D.Q. ran the experiments. Y.Q. and F.C. provided the computation resources. L.-C.X. wrote the manuscript and prepared figures with input from all the authors.

Competing interests

The authors declare no competing interests.

Reference

1. Zhang, O. *et al.* Graph Neural Networks in Modern AI-Aided Drug Discovery. *Chem. Rev.* **125**, 10001–10103 (2025).

2. Xin, H. *et al.* Roadmap for transforming heterogeneous catalysis with artificial intelligence. *Nat. Catal.* **9**, 102–111 (2026).

3. Cheng, M. *et al.* Artificial intelligence-driven approaches for materials design and discovery. *Nat. Mater.* **25**, 174–190 (2026).

4. Zollner, M., Moshfeghi, Y. & Nematiaram, T. Artificial intelligence in the discovery and design of molecular semiconductors: a systematic review. *Digit. Discov.* **5**, 1037–1067 (2026).

5. Zhang, S.-Q. & Hong, X. Mechanism and Selectivity Control in Ni- and Pd-Catalyzed Cross-Couplings Involving Carbon–Oxygen Bond Activation. *Acc. Chem. Res.* **54**, 2158–2171 (2021).

6. Ilnicka, A. & Schneider, G. Designing molecules with autoencoder networks. *Nat. Comput. Sci.* **3**, 922–933 (2023).

7. Zhu, K., Spring, D. R., Shi, B.-F. & Zhang, F. Recent progress towards catalytic asymmetric construction of inherently chiral scaffolds. *Chem. Soc. Rev.* **54**, 10856–10879 (2025).

8. Fang, Y., Pan, X. & Shen, H.-B. A comprehensive review of artificial intelligence in drug design: methods, applications, and challenges. *Digit. Discov.* 10.1039.D5DD00549C (2026) doi:10.1039/D5DD00549C.

9. Oliveira, J. C. A. *et al.* When machine learning meets molecular synthesis. *Trends Chem.* **4**, 863–885 (2022).

10. Zhu, Y. *et al.* Generative AI for controllable protein sequence design: A survey. *Npj Drug Discov.* **3**, 19 (2026).

11. Xu, L. *et al.* Transforming Molecular Synthesis With Large Language Models. *Chem. – Eur. J.* e71074 (2026) doi:10.1002/chem.71074.

12. Yin, J. *et al.* SurFF: a foundation model for surface exposure and morphology across intermetallic crystals. *Nat. Comput. Sci.* **5**, 782–792 (2025).

13. Zhao, D. *et al.* Revisiting ADMET prediction reliability under real-world challenges in the foundation model era. *J. Cheminformatics* **18**, 95 (2026).

14. Hu, R. *et al.* LamNet: an alchemical-path-aware graph neural network to accelerate binding free energy calculations for drug discovery and beyond. *Natl. Sci. Rev.* **13**, nwaf559 (2026).

15. Xu, L. *et al.* Towards Data-Driven Design of Asymmetric Hydrogenation of Olefins: Database and Hierarchical Learning. *Angew. Chem. Int. Ed.* **60**, 22804–22811 (2021).

16. Xu, L.-C. *et al.* Enantioselectivity prediction of pallada-electrocatalysed C–H activation using transition state knowledge in machine learning. *Nat. Synth.* **2**, 321–330 (2023).

17. Deng, Y. *et al.* RSGPT: a generative transformer model for retrosynthesis planning pre-trained on ten billion datapoints. *Nat. Commun.* **16**, 7012 (2025).

18. Xu, L.-C., Tang, M.-J., An, J., Cao, F. & Qi, Y. A unified pre-trained deep learning framework for cross-task reaction performance prediction and synthesis planning. *Nat. Mach. Intell.* **7**, 1561–1571 (2025).

19. Zhu, Y. *et al.* SynGFN: learning across chemical space with generative flow-based molecular discovery. *Nat. Comput. Sci*. **6**, 29–38 (2025).

20. Zhang, O. *et al.* ODesign: A World Model for Biomolecular Interaction Design. Preprint at https://doi.org/10.48550/arXiv.2510.22304 (2025).

21. Xu, L.-C., Cao, F. & Qi, Y. Stereoelectronic-aware catalyst embeddings from 2D graphs *via* 2D–3D multi-view alignment. *Digit. Discov.* 10.1039.D6DD00089D (2026) doi:10.1039/D6DD00089D.

22. Wang, J. *et al.* Token-Mol 1.0: tokenized drug design with large language models. *Nat. Commun.* **16**, 4416 (2025).

23. Fang, J. *et al.* DGMM: A Deep Learning-Genetic Algorithm Framework for Efficient Lead Optimization in Drug Discovery. *J. Chem. Inf. Model.* **65**, 8168–8180 (2025).

24. Kengkanna, A., Kikuchi, Y., Niwa, T. & Ohue, M. Reaction-conditioned generative model for catalyst design and optimization with CatDRX. *Commun. Chem.* **8**, 314 (2025).

25. Xu, L.-C. *et al.* A unified framework for automated transition state generation to accelerate mechanistic exploration in organic synthesis. Preprint at https://doi.org/10.26434/chemrxiv.10001667/v2 (2026).

26. Yao, Z. *et al.* Inverse design of nanoporous crystalline reticular materials with deep generative models. *Nat. Mach. Intell.* **3**, 76–86 (2021).

27. Reed, S. *et al.* Generative Adversarial Text to Image Synthesis. in *Proceedings of The 33rd International Conference on Machine Learning* (eds Balcan, M. F. & Weinberger, K. Q.) vol. 48 1060–1069 (PMLR, New York, New York, USA, 2016).

28. Ramesh, A. *et al.* Zero-Shot Text-to-Image Generation. in *Proceedings of the 38th International Conference on Machine Learning* (eds Meila, M. & Zhang, T.) vol. 139 8821–8831 (PMLR, 2021).

29. Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 10674–10685 (2022). doi:10.1109/CVPR52688.2022.01042.

30. Singer, U. *et al.* Make-A-Video: Text-to-Video Generation without Text-Video Data. Preprint at https://doi.org/10.48550/arXiv.2209.14792 (2022).

31. Ho, J. *et al.* Video Diffusion Models. in *Advances in Neural Information Processing Systems* (eds Koyejo, S. et al.) vol. 35 8633–8646 (Curran Associates, Inc., 2022).

32. Hong, W., Ding, M., Zheng, W., Liu, X. & Tang, J. CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers. in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023* (OpenReview.net, 2023).

33. Wu, Z. *et al.* Leveraging language model for advanced multiproperty molecular optimization via prompt engineering. *Nat. Mach. Intell.* **6**, 1359–1369 (2024).

34. Cavanagh, J. M. *et al.* SmileyLlama: modifying large language models for directed chemical space exploration. *Nat. Comput. Sci.* https://doi.org/10.1038/s43588-026-00986-y (2026) doi:10.1038/s43588-026-00986-y.

35. Edwards, C. *et al.* Translation between Molecules and Natural Language. in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* 375–413 (Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022). doi:10.18653/v1/2022.emnlp-main.26.

36. Gong, H., Liu, Q., Wu, S. & Wang, L. Text-Guided Molecule Generation with Diffusion Language Model. *Proc. AAAI Conf. Artif. Intell.* **38**, 109–117 (2024).

37. Zhou, P. *et al.* Instruction multi-constraint molecular generation using a teacher-student large language model. *BMC Biol.* **23**, 105 (2025).

38. Xiang, Y., Zhao, H., Ma, C. & Deng, Z.-H. Instruction-Based Molecular Graph Generation with Unified Text-Graph Diffusion Model. Preprint at https://doi.org/10.48550/arXiv.2408.09896 (2025).

39. Liu, S. *et al.* Multi-modal molecule structure–text model for text-based retrieval and editing. *Nat. Mach. Intell.* **5**, 1447–1457 (2023).

40. Tingle, B. I. *et al.* ZINC-22─A Free Multi-Billion-Scale Database of Tangible Compounds for Ligand Discovery. *J. Chem. Inf. Model.* **63**, 1166–1176 (2023).

41. Lipinski, C. A., Lombardo, F., Dominy, B. W. & Feeney, P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Deliv. Rev.* **23**, 3–25 (1997).

42. Yang, A. *et al.* Qwen3 Technical Report. Preprint at https://doi.org/10.48550/arXiv.2505.09388 (2025).

43. Radford, A. *et al.* Learning Transferable Visual Models From Natural Language Supervision. in *Proceedings of the 38th International Conference on Machine Learning* (eds Meila, M. & Zhang, T.) vol. 139 8748–8763 (PMLR, 2021).

44. Gensch, T. *et al.* A Comprehensive Discovery Platform for Organophosphorus Ligands for Catalysis. *J. Am. Chem. Soc.* **144**, 1205–1217 (2022).

45. Singh, A. *et al.* OpenAI GPT-5 System Card. Preprint at https://doi.org/10.48550/arXiv.2601.03267 (2026).

46. DeepSeek-AI *et al.* DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence. Preprint at https://doi.org/10.48550/arXiv.2606.19348 (2026).