MOSA: MOLECULAR SUBSPACE ASSEMBLY FOR UNIFIED MOLECULAR GENERATION AND EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular design often requires both generating new molecules from multiple property specifications and editing existing molecules while preserving their underlying structures. Although these tasks are closely related, conventional joint adaptation can benefit one task while degrading the other. We introduce **MOSA (Molecular Subspace Assembly)**, a unified adapter for a frozen language model that organizes low-rank updates into shared, generation-private, and editing-private subspaces. A permutation-invariant property controller uses individual conditions and their pairwise interactions to modulate the shared update. For editing, a source controller combines the input molecule with the requested properties to scale and shift the editing-specific update. The same frozen backbone and text interface serve both tasks, while each request changes the adaptation applied to hidden states. At 39,500 updates, MOSA achieves **60.45%** strict de novo success on 6,000 common requests versus SketchMol's 54.5% under native training budgets, and **76.98%** strict editing success on 5,000 requests. At matched one-pass exposure, MOSA improves both tasks over a conventional joint adapter and avoids the editing-side negative transfer observed relative to the editing specialist.

1 INTRODUCTION

Molecular design involves both generating new molecules with desired properties and modifying existing molecules while preserving their underlying structures. De novo generation starts without a source molecule and seeks an output that satisfies the requested properties (Liu et al., 2024a). Molecular editing instead starts from an existing molecule and must satisfy the requested changes while remaining close to the source (Zhuang et al., 2026).

Multi-property molecular design requires several objectives to be satisfied simultaneously (Fromer & Coley, 2023). For a de novo generation request, many different molecules may satisfy the same property targets. An editing request is more restrictive because property satisfaction alone is not sufficient. An output may satisfy all requested properties yet still fail as an edit if it deviates too far from the source. Figure 1 illustrates this distinction.

Molecular generators use graphs, sequences, diffusion, and multimodal representations (Jin et al., 2018; Shi et al., 2020; Liu et al., 2024a; Fang et al., 2024b). Editing methods condition a change on a supplied molecule (Liu et al., 2023; 2024b; Ye et al., 2025; Zhuang et al., 2026). More recent models bring molecular structures, properties, language, and images into shared interfaces (Fang et al., 2024a; Tang et al., 2024; Chang & Ye, 2024; Wang et al., 2025). A shared interface lets one model receive both kinds of requests, but it does not determine how the two tasks should share trainable parameters. That is the question we study.

Joint training can transfer useful knowledge, but it can also cause *negative transfer*: a task performs worse than a model trained only on that task after seeing the same task examples. Both generation and editing need property control, while editing must also retain its source. With one shared adapter, losses from both tasks update the same parameters despite these different constraints. Separate adapters avoid cross-task updates but give up joint learning. Shared/private representations and conditional adapters offer a middle ground (Bousmalis et al., 2016; Hu et al., 2022; Yang et al., 2025; Feng et al., 2024; Perez et al., 2018). Their use here raises a concrete design question: which update should respond to the property request, and which should also respond to the source?

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.

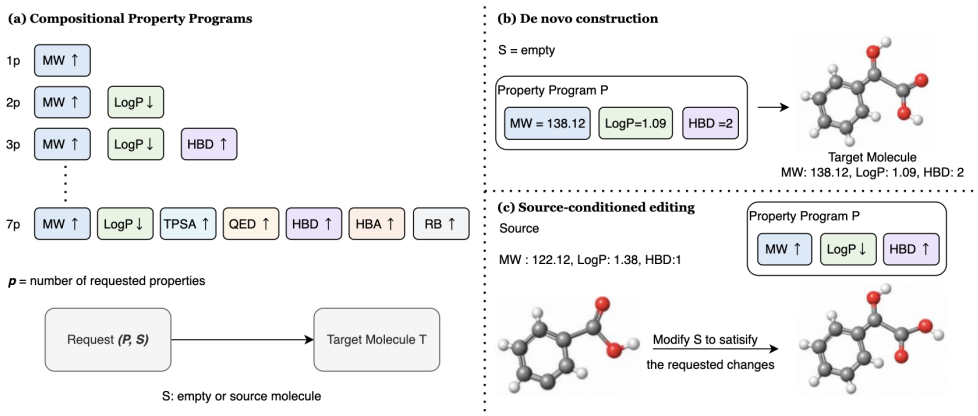


Figure 1: Unified molecular design. (a) A program combines p property conditions. (b) Generation uses numerical targets and an empty source. (c) Editing adds a source and a structural-retention requirement. The illustrated program P corresponds to C in Section 3. Molecules and values illustrate task semantics, not measured outputs.

Two aspects of molecular requests make parameter sharing more difficult. First, a request may contain several property conditions, and both the number and combination of these conditions vary across requests. The model must satisfy them jointly rather than treating each condition in isolation. This motivates a representation that can capture both individual property goals and their interactions. Second, molecular editing is tied to a specific source molecule. The same property request may require different structural changes for different starting molecules. Although the source is included in the text input, this alone does not make the adapter parameters explicitly depend on the source.

These differences also affect evaluation. Generation must satisfy the property goals, while editing must also preserve the source structure. We therefore assess the two tasks separately rather than let improvement on one hide a loss on the other.

We introduce **MOSA (Molecular Subspace Assembly)** to address this sharing problem. A frozen language model has one shared low-rank path and one private path for each task. The requested properties modulate the shared path. For editing, the source molecule and the properties also modulate its private path. The generation path does not require a source. A single backbone and text interface handle both tasks, while each request changes the update applied inside the model (Section 3). This selective sharing is designed to mitigate negative transfer: both tasks can learn property control through the shared path, while each private path receives completion-loss gradients from only its own task.

We evaluate whether this design improves both tasks under matched own-task exposure, then test the capacity and conditioning choices separately. We also compare task-level performance with published molecular design methods. The experiments distinguish a gain from joint learning from a gain caused only by storing more adapter parameters. Code and evaluation details are available anonymously.[1]

Our contributions are threefold:

- **Negative transfer under matched own-task exposure.** We show that conventional joint adaptation improves de novo generation but degrades molecular editing relative to their single-task specialists when task exposure is matched.
- **Selective molecular adaptation.** We introduce MOSA, which combines a property-conditioned shared update with separate task-specific updates. The editing-specific update is further conditioned on the source molecule.
- **Controlled evaluation.** We evaluate MOSA on strict de novo generation and source-preserving editing, and test its design with matched-capacity, learned-composition, and source-modulation controls.

[1] https://anonymous.4open.science/r/MOSA-60C6/

2 Related Work

Molecular generation and editing. Molecular generation spans latent graphs and motifs (Jin et al., 2018; 2020), autoregressive models (Bagal et al., 2022; Shi et al., 2020), and diffusion (Vignac et al., 2023; Liu et al., 2024a); see the broader surveys (Meyers et al., 2021; Mouchlis et al., 2021). MolT5 and MolGen connect generation with language and chemical feedback (Edwards et al., 2022; Fang et al., 2024b). MoleculeSTM aligns molecules with text for editing (Liu et al., 2023); ChatDrug, DrugAssist and GeLLM^{3}O support requested changes through retrieval, interaction or multi-property optimization (Liu et al., 2024b; Ye et al., 2025; Dey et al., 2025). REINVENT 4 covers several design modes (Loeffler et al., 2024), and MolEditRL uses graph diffusion with reinforcement learning for structure-preserving edits (Zhuang et al., 2026). Together, these studies establish the design settings and task-specific approaches. We ask how construction and source-conditioned editing should share one adapted backbone.

Unified molecular models. Mol Instructions provides instruction data across biomolecular tasks (Fang et al., 2024a). MolLM integrates text with molecular structures (Tang et al., 2024). SPMM supports bidirectional structure–property generation (Chang & Ye, 2024), while SketchMol uses images for de novo design and masked editing (Wang et al., 2025). These works broaden the supported tasks or representations. Our focus is the adaptation mechanism that lets one backbone serve both construction and editing.

Multi-task and conditional adaptation. LoRA and AdaLoRA add low-rank updates to frozen models (Hu et al., 2022; Zhang et al., 2023). Shared/private learning predates this setting (Bousmalis et al., 2016); MTL-LoRA and mixtures add task-adaptive parameters or route among adapters (Yang et al., 2025; Feng et al., 2024). FiLM and hypernetworks condition features or weights on context (Perez et al., 2018; Ha et al., 2017). MOSA assigns property-controlled directions to both tasks, but source-and-property-controlled directions only to editing. This differs from selecting a static adapter by task. We evaluate the complete model separately from individual modulation choices.

3 Method

3.1 Allocating updates across two tasks

A request specifies a set of property goals C and, for editing, a source molecule S. Both tasks use the same frozen transformer and predict a plan label followed by a target SMILES. Their update requirements differ: generation must construct a molecule satisfying C, whereas editing must satisfy C while retaining information from S. With an ordinary joint LoRA adapter, both completion losses update the same low-rank factors at every adapted projection (Hu et al., 2022). MOSA instead reserves one low-rank path for both tasks and one path for each task. Both tasks train the property-responsive shared path; each private path receives completion-loss gradients only from its own task. This confines direct cross-task completion-loss updates to the shared path while retaining a route for useful transfer.

3.2 Request-conditioned subspace assembly

At projection ℓ, let h be the incoming transformer state and W_ℓ the frozen weight. We partition the trainable LoRA factors into shared (c), generation-private (d), and editing-private (e) blocks: $A_\ell = [A_c^\ell; A_d^\ell; A_e^\ell]$ and $B_\ell = [B_c^\ell\ B_d^\ell\ B_e^\ell]$. Each block has rank eight. The three candidate updates are

$$\begin{aligned} u_c^\ell(h, C) &= B_c^\ell\big(g_c^\ell(C) \odot A_c^\ell h\big), \\ u_d^\ell(h) &= B_d^\ell A_d^\ell h, \\ u_e^\ell(h, C, S) &= B_e^\ell\big(g_e^\ell(C, S) \odot A_e^\ell h + b_e^\ell(C, S)\big). \end{aligned} \tag{1}$$

The coefficient vectors g_c^ℓ, g_e^ℓ, and b_e^ℓ each have eight entries. The property controller supplies g_c^ℓ; a source-and-property controller supplies the editing scale g_e^ℓ and shift b_e^ℓ. With the registered LoRA

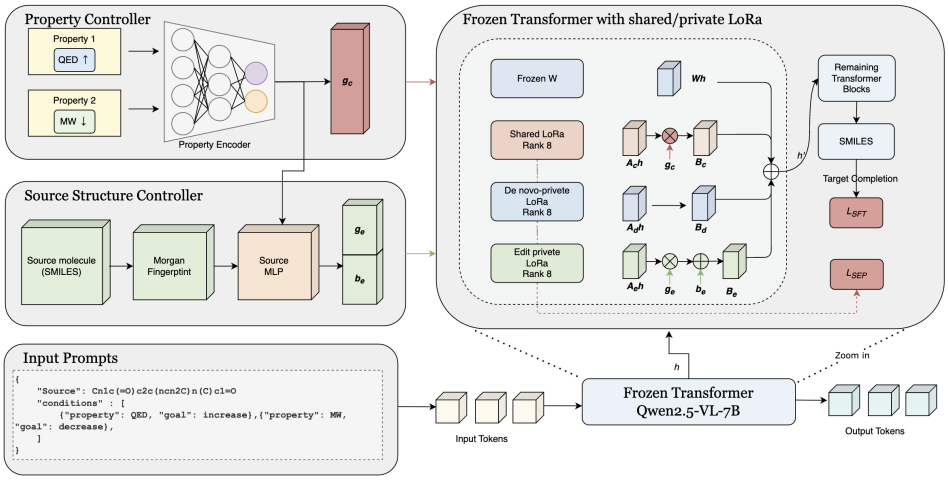


Figure 2: MOSA at an adapted projection. The property controller scales the shared rank block. Generation uses its private block; editing uses a source-and-property-conditioned private block with scale and shift. The backbone remains frozen.

scaling of two, MOSA assembles the projection output as

$$h'_\ell = \begin{cases} W_\ell h + 2\big(u_c^\ell(h, C) + u_d^\ell(h)\big), & \text{generation,} \\ W_\ell h + 2\big(u_c^\ell(h, C) + u_e^\ell(h, C, S)\big), & \text{editing.} \end{cases} \tag{2}$$

Each request uses the shared block and one private block, activating 16 of the 24 stored rank directions. Both tasks train the shared block; only generation or editing supplies completion-loss gradients to its respective private block. Unlike a fixed task route, the coefficients within the active blocks can change with the requested properties and, for editing, the source. These coefficients act on LoRA rank features rather than on slices of h. The source tokens may also change h itself, so a fixed C does not imply an identical shared output for different sources.

3.3 Controllers and Joint Training

Property program. The property controller encodes each property-goal pair, then combines contributions from individual conditions with a symmetric summary of pairwise interactions. Projection-specific heads produce $g_c^\ell(C)$. The interaction summary allows a conjunction of goals to affect the shared update beyond the separate goals. Because the controller aggregates a set, its coefficients are unchanged by permuting the conditions; the transformer still reads their serialized prompt in token order. Appendix B gives the exact encoding and coefficient map.

Source-conditioned editing. A Morgan fingerprint of the input molecule is encoded together with the property summary. A head for each projection yields $g_e^\ell(C, S)$ and $b_e^\ell(C, S)$. The source SMILES also appears in the transformer prompt, but only this auxiliary path directly changes the editing-private coefficients. It never uses the target molecule. Coefficients are computed once per request and reused during decoding. Appendix B gives their dimensions and ranges.

Objective. We jointly train the LoRA factors and both controllers with completion-token cross entropy $\mathcal{L}_{\text{SFT}}$. Prompt and padding tokens are masked. A separation penalty discourages alignment between the shared and private down-projection rows:

$$\mathcal{L} = \mathcal{L}_{\text{SFT}} + \lambda_t \operatorname*{mean}_{\ell,\, q \in \{d,e\}} \frac{\big\| \widehat{A}_c^\ell (\widehat{A}_q^\ell)^\top \big\|_F^2}{8^2}, \tag{3}$$

where the rows of $\widehat{A}$ have unit norm. The penalty acts on weight-space geometry, while the completion objective determines what each block learns. The LoRA up projections start at zero, controller scales

at one, and shifts at zero, so training starts from the frozen backbone. Appendix B gives the penalty schedule and implementation details.

4 EXPERIMENTS

4.1 TASKS AND DATA

Each request contains a property program and, for editing, a source SMILES. Generation programs request two to seven properties and editing programs request one to seven. Programs use numerical targets or, when a source is provided, directions relative to it. The training pool contains 2,000,000 unique generation targets derived from PubChem structures (Kim et al., 2025) and 569,919 unique source–target editing pairs from the MolEdit-Instruct training split (Zhuang et al., 2026). Training targets never appear in the input request.

The one-pass training stream selects a balanced subset of these pools without replacement. It draws 81,415 records from each of six generation and seven editing property-count buckets, yielding 488,490 generation and 569,905 editing records. Programs use MW, LogP, QED, TPSA, HBD, HBA, and rotatable bonds. The ten-task editing evaluation also includes SA and predicted GSK3B/DRD2 activity. Appendix A gives the request format, property tolerances, oracle definitions, and data checks.

4.2 EVALUATION METRICS

The complete generation panel has 6,000 requests, with 1,000 for each property count from two to seven. Editing has 5,000 requests, with 500 sources for each of ten tasks. Strict generation success requires a valid molecule that satisfies *every* requested property. Strict editing success also requires radius-two, 2,048-bit Morgan-fingerprint Tanimoto similarity of at least 0.65 to the source. We report the fraction of all requests that pass. Invalid and missing outputs remain in the denominator. Validity is reported separately; relaxed editing success uses a similarity threshold of 0.15 where shown.

Raw@1 means one sampled completion per request, without repair or reranking. Internal models use temperature 0.8, top-p 0.95, and top-k 50. The same requests and decoding rule are used within internal paired comparisons. Appendix A specifies the exact property tolerances, editing tasks, and remaining decoding settings.

4.3 BASELINES

Generation baselines are SPMM (Chang & Ye, 2024), PEIT-LLM (Lin et al., 2024), and Sketch-Mol (Wang et al., 2025). Editing public references are BioT5 (Pei et al., 2023), MolGen (Fang et al., 2024b), REINVENT4 (Loeffler et al., 2024), DrugAssist (Ye et al., 2025), GeLLM^{3}O (Dey et al., 2025), and MolEditRL (Zhuang et al., 2026).

For internal comparisons, a generation specialist and an editing specialist establish single-task performance. Joint uses one rank-16 adapter for both tasks. Joint-24 increases its stored LoRA rank to match MOSA's LoRA parameter count. MOSA-Fixed retains the three-branch layout with controller scales fixed to one and shifts fixed to zero. A retrained no-source-modulation arm leaves the source SMILES in the prompt but fixes the editing-private source coefficients. These controls separate joint transfer, stored capacity, learned composition, and source modulation.

4.4 TRAINING AND COMPARISON PROTOCOL

All internal models freeze Qwen2.5-VL-7B-Instruct (Bai et al., 2025) and use effective batch 65. Specialists train on their own task records; joint models see both tasks. MOSA and MOSA-Fixed store three rank-eight branches, with 16 directions active per request. Table 1 lists training populations and checkpoints.

The one-pass comparison matches examples seen for each task, not total updates or compute: specialists reach 7,516 generation or 8,768 editing updates, while joint models reach 16,283 updates. We evaluate these models on the complete 6,000/5,000 panels. Three independent MOSA runs use this one-pass schedule. Separately, MOSA and MOSA-Fixed are compared at equal updates on the same 600-generation/500-editing development requests and on the complete panels at update 39,500.

External comparisons use MOSA at that extended checkpoint. Appendix B gives the exposure calculation and Appendix E.1 documents the continuation.

Table 1: Internal training populations and evaluated checkpoints. Record counts are for the first pass; the effective batch is 65. A dash indicates that a specialist does not train on that task.

Model	Construction	Editing	Adapter	Evaluated updates
De novo specialist	488,490	–	LoRA $r = 16$	7,516
Editing specialist	–	569,905	LoRA $r = 16$	8,768
Joint	488,490	569,905	LoRA $r = 16$	16,283
Joint-24	488,490	569,905	LoRA $r = 24$	16,283
MOSA-Fixed	488,490	569,905	$8 + 8 + 8$, fixed	16,283 / 28,496 / 39,500
MOSA	488,490	569,905	$8 + 8 + 8$, learned	16,283 / 28,496 / 39,500

5 RESULTS

5.1 MATCHED COMPARISON REVEALS EDITING-SIDE NEGATIVE TRANSFER

Table 2 compares joint and single-task adaptation at the full one-pass endpoint. Relative to the task specialists, Joint gains 4.1 generation points but loses 4.5 editing points. Under matched own-task exposure, this is editing-side negative transfer. MOSA improves both tasks, by 8.5 generation and 2.5 editing points, and avoids the editing regression. It also exceeds Joint and the generation specialist at every property count from two to seven (Appendix D.1). Appendix Figure 5 traces the three methods on common development requests across training.

Table 2: Matched one-pass comparison at equal own-task exposure. Strict success uses the complete 6,000-generation/5,000-editing panels. Δ is the change from the corresponding specialist in percentage points, computed before rounding. Specialist denotes separate models for the two tasks.

	De novo (6,000)		Editing (5,000)	
Model	Strict (%)	Δ (pp)	Strict (%)	Δ (pp)
Specialist	44.2	–	69.0	–
Joint	48.3	$+4.1$	64.5	-4.5
MOSA	**52.7**	$\mathbf{+8.5}$	**71.5**	$\mathbf{+2.5}$

Paired uncertainty. All four paired 95% intervals for MOSA's gains over Joint and the specialists lie above zero (Table 3). Appendix D.2 gives the cluster-bootstrap procedure.

Table 3: MOSA minus the one-pass baselines in percentage points, with pointwise 95% paired cluster-bootstrap intervals conditional on the trained models.

Panel	Comparison	Difference	95% interval
De novo	MOSA – Specialist	+8.53	[+7.02, +9.96]
De novo	MOSA – Joint	+4.42	[+2.96, +5.88]
Editing	MOSA – Specialist	+2.54	[+1.44, +3.65]
Editing	MOSA – Joint	+7.02	[+5.67, +8.32]

Across three independent MOSA training seeds, generation and editing success average $53.49 \pm 0.89\%$ and $71.31 \pm 0.47\%$, respectively (sample SD). Appendix Table 20 gives each run.

5.2 EXTERNAL MOLECULAR DESIGN COMPARISONS

External benchmarks use request-conditioned **MOSA** at update 39,500 on complete panels. This continuation is distinct from the one-pass matched model (Appendix E.1).

Molecular editing. On the complete 5,000-request editing panel, MOSA reaches 76.98% strict success. Table 4 compares results by requested property count with published public references, including MolEditRL's 45.0% overall value. The ten task-wise values are in Appendix Table 27, with provenance in Appendix G.1.

Table 4: External editing comparison by number of requested properties (p), in percent. Strict success uses similarity ≥ 0.65, and relaxed success uses ≥ 0.15. MOSA uses the 39,500-update checkpoint on 5,000 requests. Baseline values are published public references.

	Strict $A_{.65}$				Auxiliary	
Method	1p	2p	3p	Overall	$A_{.15}$	Valid.
BioT5	0.0	0.0	0.0	0.0	29.2	**100.0**
MolGen	2.4	3.9	3.6	3.2	22.2	**100.0**
REINVENT4	16.0	12.6	5.5	12.5	25.0	64.1
DrugAssist	38.4	32.5	32.6	34.9	41.1	97.9
GeLLM^{3}O-M	22.3	13.6	4.5	15.2	35.4	90.2
GeLLM^{3}O-L	18.1	14.6	7.4	14.5	40.4	90.1
MolEditRL	50.2	38.7	47.4	45.0	72.7	96.6
MOSA	**73.6**	**78.5**	**80.7**	**77.0**	**89.4**	97.7

De novo generation. On the common 6,000-request panel, MOSA reaches 60.45% strict success versus SketchMol's 54.5% and SPMM's 45.6% (Table 5). It leads SketchMol in all six property-count groups. These methods use their native training budgets and representations, so this comparison measures benchmark performance rather than the effect of MOSA's controller.

Table 5: External de novo comparison on the common 6,000-request Raw@1 panel. Columns 2p–7p report strict success by number of requested properties. External methods appear first. MOSA is evaluated at update 39,500 (Appendix E.1).

Method	2p	3p	4p	5p	6p	7p	Mean	Valid.
SPMM	60.2	52.6	45.2	39.4	39.7	36.6	45.6	**97.6**
PEIT-LLM	40.2	24.9	21.0	12.9	10.5	6.9	19.4	93.9
SketchMol	74.1	67.8	56.5	47.3	43.7	37.6	54.5	–
MOSA	**76.2**	**70.3**	**63.3**	**56.8**	**50.0**	**46.1**	**60.5**	97.1

Property complexity. At seven properties, the extended MOSA checkpoint retains 46.1% success versus SketchMol's 37.6% and SPMM's 36.6%.

5.3 Where editing-side negative transfer appears

The editing-side regression is visible in the failure categories on 5,000 identical requests. Joint has 3,225 strict successes versus 3,449 for the editing specialist and 3,576 for MOSA. It produces 1,032 outputs that satisfy the properties but miss the structural threshold, versus 855 for the specialist and 771 for MOSA. This analysis localizes much of the observed editing gap to structural retention, but does not identify the optimization mechanism that causes negative transfer. Figure 3 shows that MOSA's advantage over Joint does not persist at similarity thresholds 0.8–0.9.

5.4 Architecture ablations

Table 6 shows complete-panel results for three retrained controls. Capacity and source controls use one pass; the composition control uses 39,500 updates. Requests and updates match within each block.

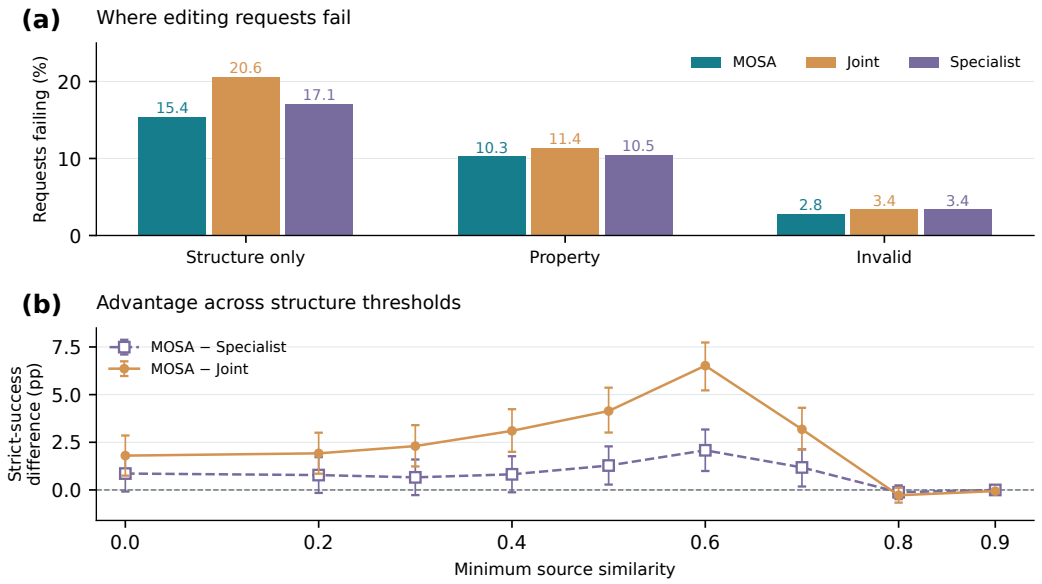


Figure 3: Editing outcomes on 5,000 identical requests at one pass. (a) Exclusive failures at similarity 0.65. (b) MOSA minus each baseline across similarity thresholds, with pointwise paired 95% source-cluster intervals. The MOSA–Joint advantage does not persist at 0.8–0.9. Appendix F gives the full rates.

Table 6: Architecture controls on the complete 6,000-generation/5,000-editing panels. Strict success is in percent. Models within each block use the same requests and update count; the blocks are separate training endpoints. Joint gives the ordinary shared-adapter reference.

Model	Updates	De novo	Editing	Comparison role
One-pass controls				
Joint	16,283	48.28	64.50	Ordinary joint adapter
Joint-24	16,283	52.48	66.82	Stored LoRA capacity
No-source modulation	16,283	**53.07**	70.06	Source controller
MOSA	16,283	52.70	**71.52**	Full model
Extended composition comparison				
MOSA-Fixed	39,500	59.40	76.22	Fixed coefficients
MOSA	39,500	**60.45**	**76.98**	Learned coefficients

Stored adapter capacity. Joint-24 matches MOSA's stored LoRA capacity. Generation differs by +0.22 points (95% paired interval $[-1.25, 1.71]$), while MOSA leads editing by +4.70 $[3.46, 5.90]$. Stored rank alone does not explain this gap. MOSA also adds 377,984 controller parameters and differs in active rank and branch design (Appendix E.3).

Learned composition. MOSA-Fixed keeps the shared/private branches but fixes controller coefficients. At 39,500 updates, MOSA leads by +1.05 generation points (95% paired interval $[-0.23$, +2.30]) and +0.76 editing points. The development trajectory in Appendix Table 21 informed checkpoint selection. Because controller parameters and compute also differ, the complete-panel contrast does not isolate coefficient variation.

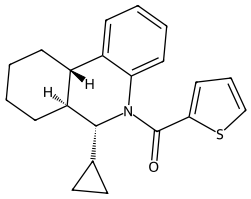
Source modulation. Removing source modulation during training lowers editing success by 1.46 points (95% paired source-cluster interval [0.58, 2.38]); generation is similar. Source SMILES stays in the prompt, but editing-private scales and shifts are fixed. Further controls appear in Appendix E.2.

5.5 Molecular examples

Figure 4 shows deterministically selected outputs with recomputed descriptors. Further cases and the selection protocol appear in Appendix F.2.

The generated molecule satisfies all seven constraints under the stated tolerances. In the editing example, MW rises from 446.38 to 564.54 and HBA from 3 to 4, while QED falls from 0.297 to 0.189. Source similarity is 0.743, above the 0.65 threshold.

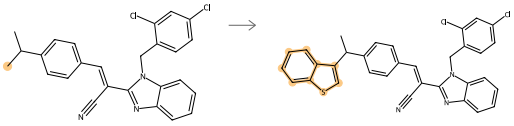
(a) 7-property generation



Property	Target	Output
MW	334.55	**337.49**
LogP	4.95	**5.46**
QED	0.734	**0.712**
TPSA	20.23	**20.31**
HBD	1	**0**
HBA	3	**2**
RB	1	**2**

(b) 3-property editing

Source → Generated



Request	Source	Output
MW up	446.38	**564.54**
QED down	0.297	**0.189**
HBA up	3	**4**

Source similarity 0.743

Figure 4: Saved Raw@1 MOSA outputs at the matched endpoint: (a) seven-property generation and (b) source-conditioned editing. All displayed requests pass. Orange marks atoms outside an exact-atom maximum common substructure. Appendix F.2 gives further cases.

6 Discussion and Limitations

At matched own-task exposure, Joint shows asymmetric transfer: generation improves over its specialist, while editing degrades. The additional editing failures are concentrated among outputs that satisfy the requested properties but fall below the source-similarity threshold, suggesting that the main difficulty is balancing property control with source preservation. MOSA reduces this trade-off through selective sharing: both tasks use a shared property-related update, while task-specific updates remain private and editing is conditioned on the source. Joint-24 stores the same LoRA capacity but still performs worse on editing, suggesting that stored rank alone does not explain the improvement. Our results establish negative transfer at the task-performance level, but do not identify its optimization mechanism.

Our evaluation has several limitations. Morgan-fingerprint similarity is only a proxy for structural preservation and does not guarantee scaffold, stereochemical, or geometric similarity. The experiments use fixed-format requests and directional editing objectives, and do not establish molecule- or scaffold-disjoint generalization. Descriptor and activity-predictor scores also do not establish synthesizability, safety, or biological efficacy.

7 Conclusion

Conventional joint adaptation benefits de novo generation but introduces negative transfer on molecular editing. MOSA uses selective shared and task-specific adaptation to improve both tasks under matched exposure, suggesting that selective parameter sharing can unify generation and editing while preserving their distinct requirements.

AI USE STATEMENT

Generative AI tools were used as a supporting tool during the preparation of this work. The research question, proposed method, experimental design, and scientific conclusions were developed by the authors. AI tools were used to discuss and critically examine ideas developed by the authors, and to improve the clarity and presentation of the manuscript. All experiments, results, references, figures, and claims were verified by the authors, who remain fully responsible for the content of the paper.

ETHICS STATEMENT

This work is a computational study of molecular generation and editing. It does not involve human participants or personal data. The generated molecules are evaluated using computational descriptors, activity predictors, and structural similarity measures. These evaluations do not establish that a molecule is synthesizable, safe, or biologically effective.

Molecular generation models may also have dual-use implications. The methods and outputs in this work are intended for research purposes. Any molecule proposed by the model should be subject to appropriate expert review and experimental validation before practical use.

REPRODUCIBILITY STATEMENT

We provide the model architecture, training procedure, evaluation protocol, hyperparameters, decoding settings, checkpoint definitions, and random seeds in the main paper and appendices. The paper also reports the construction of the training streams, the task-matched exposure protocol, evaluation metrics, and statistical procedures.

An anonymous implementation is provided with the submission. It contains the method, training and evaluation code, main configurations, and implementation tests. Datasets, evaluation panels, model weights, and third-party activity predictors are not bundled with the repository. The repository documents the required external artifacts and the intended reproduction scope.

REFERENCES

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain Separation Networks. In *Advances in Neural Information Processing Systems*, 2016. URL https://arxiv.org/abs/1608.06019.

Jinho Chang and Jong Chul Ye. Bidirectional generation of structure and properties through a single molecular foundation model. *Nature Communications*, 15:2323, 2024. doi: 10.1038/s41467-024-46440-3.

Vishal Dey, Xiao Hu, and Xia Ning. GeLLM^{3}O: Generalizing large language models for multi-property molecule optimization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 25192–25221, 2025. doi: 10.18653/v1/2025.acl-long.1225. URL https://aclanthology.org/2025.acl-long.1225/.

Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between Molecules and Natural Language. In *EMNLP*, 2022. URL https://aclanthology.org/2022.emnlp-main.26/.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. In *International Conference on Learning Representations*, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/d346d91999074dd8d6073d4c3b13733b-Abstract-Conference.html.

Yin Fang, Ningyu Zhang, Zhuo Chen, Lingbing Guo, Xiaohui Fan, and Huajun Chen. Domain-agnostic molecular generation with chemical feedback. In *International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=9rPyHyjfwP.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. Mixture-of-LoRAs: An Efficient Multitask Tuning Method for Large Language Models. In *LREC-COLING*, 2024. URL https://aclanthology.org/2024.lrec-main.994/.

Jenna C. Fromer and Connor W. Coley. Computer-aided multi-objective optimization in small molecule discovery. *Patterns*, 2023. URL https://doi.org/10.1016/j.patter.2023.100678.

David Ha, Andrew Dai, and Quoc V. Le. HyperNetworks. In *ICLR*, 2017. URL https://arxiv.org/abs/1609.09106.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*, 2022. URL https://arxiv.org/abs/2106.09685.

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

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4839–4848, 2020. URL https://proceedings.mlr.press/v119/jin20a.html.

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

Xuan Lin, Long Chen, Yile Wang, Yangyang Chen, and Xiangxiang Zeng. Property enhanced instruction tuning for multi-task molecule generation with large language models, 2024. URL https://arxiv.org/abs/2412.18084.

Gang Liu, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph diffusion transformers for multi-conditional molecular generation. In *Advances in Neural Information Processing Systems*, 2024a. URL https://openreview.net/forum?id=cfrDLD1wfO.

Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Anima Anandkumar. Multi-modal molecule structure–text model for text-based retrieval and editing. *Nature Machine Intelligence*, 5(12):1447–1457, 2023. doi: 10.1038/s42256-023-00759-6.

Shengchao Liu, Jiongxiao Wang, Yijin Yang, Chengpeng Wang, Ling Liu, Hongyu Guo, and Chaowei Xiao. Conversational Drug Editing Using Retrieval and Domain Feedback. In *ICLR*, 2024b. URL https://github.com/chao1224/ChatDrug.

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. *Journal of Cheminformatics*, 2024. URL https://doi.org/10.1186/s13321-024-00812-5.

Joshua Meyers, Benedek Fabian, and Nathan Brown. De Novo Molecular Design and Generative Models. *Drug Discovery Today*, 2021. URL https://doi.org/10.1016/j.drudis.2021.05.019.

Varnavas D. Mouchlis, Antreas Afantitis, Angela Serra, Michele Fratello, Anastasios G. Papadiamantis, Vassilis Aidinis, Iseult Lynch, Dario Greco, and Georgia Melagraki. Advances in De Novo Drug Design: From Conventional to Machine Learning Methods. *International Journal of Molecular Sciences*, 2021. URL https://doi.org/10.3390/ijms22041676.

Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. BioT5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1102–1123, 2023. doi: 10.18653/v1/2023.emnlp-main.70.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. In *AAAI*, 2018. URL https://arxiv.org/abs/1709.07871.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. GraphAF: A Flow-based Autoregressive Model for Molecular Graph Generation. In *ICLR*, 2020. URL https://arxiv.org/abs/2001.09382.

Xiangru Tang, Andrew Tran, Jeffrey Tan, and Mark B. Gerstein. MolLM: A Unified Language Model for Integrating Biomedical Text with 2D and 3D Molecular Representations. *Bioinformatics*, 2024. URL https://doi.org/10.1093/bioinformatics/btae260.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete Denoising Diffusion for Graph Generation. In *ICLR*, 2023. URL https://arxiv.org/abs/2209.14734.

Zixu Wang, Yangyang Chen, Pengsen Ma, Zhou Yu, Jianmin Wang, Yuansheng Liu, Xiucai Ye, Tetsuya Sakurai, and Xiangxiang Zeng. Image-based generation for molecule design with sketchmol. *Nature Machine Intelligence*, 7:244–255, 2025. doi: 10.1038/s42256-025-00982-3.

Yaming Yang, Dilxat Muhtar, Yelong Shen, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Weiwei Deng, Feng Sun, Qi Zhang, Weizhu Chen, and Yunhai Tong. MTL-LoRA: Low-Rank Adaptation for Multi-Task Learning. In *AAAI*, 2025. doi: 10.1609/aaai.v39i20.35509. URL https://ojs.aaai.org/index.php/AAAI/article/view/35509.

Geyan Ye, Xibao Cai, Houtim Lai, Xing Wang, Junhong Huang, Longyue Wang, Wei Liu, and Xiangxiang Zeng. DrugAssist: A large language model for molecule optimization. *Briefings in Bioinformatics*, 26(1), 2025. doi: 10.1093/bib/bbae693.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning. In *ICLR*, 2023. URL https://arxiv.org/abs/2303.10512.

Yuanxin Zhuang, Dazhong Shen, and Ying Sun. MolEditRL: Structure-preserving molecular editing via discrete diffusion and reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=40QphlZ9fY.

APPENDIX CONTENTS

A Requests, data, and evaluation protocol . . . 14
B Architecture and training details . . . 15
C MOSA algorithm . . . 17
D One-pass checkpoints and complete-panel results . . . 18
E Design and retraining controls . . . 23
F Failure analysis and molecular examples . . . 25
G Additional evaluations . . . 28
H Limitations and Future Directions . . . 29

A REQUESTS, DATA, AND EVALUATION PROTOCOL

A request contains a source SMILES (`<EMPTY>` for de novo) and one to seven distinct property conditions. Numerical goals use {`"around": target`}. Directional goals use `increase` or `decrease`. Completions contain a `BUILD`/`MODIFY` plan label and SMILES. The controller receives only input conditions and the input-source fingerprint. Target tokens enter only the completion loss. Conditions are sorted and padding is masked in the auxiliary encoder.

Properties and success. The table gives numerical tolerances and separate encoder scales. A numerical request passes when $|f_p(y) - t| \leq \tau_p$. Editing directions require values strictly above or below those of the source. Missing required oracle values count as failures. SA and predicted GSK3B/DRD2 activity use the registered oracles. The DRD2 task requests a decrease. These are computational objectives rather than measured biological activity.

Table 7: Implemented numerical descriptor contract. Tolerances apply to `around` goals. The final column scales numerical inputs before the inverse hyperbolic sine.

Property	RDKit descriptor	Tolerance τ_p	Encoder scale s_p
MW	`Descriptors.MolWt`	35	500
LogP	`Crippen.MolLogP`	1.0	5
QED	`QED.qed`	0.10	1
TPSA	`CalcTPSA`	20	150
HBD	`NumHDonors`	1	10
HBA	`NumHAcceptors`	1	10
RB	`NumRotatableBonds`	1	10

Training population. The frozen stream has six generation buckets (two through seven properties) and seven editing buckets (one through seven). Each contributes 81,415 records. Specialists filter the same stream to their own tasks while preserving record order. Records have equal weight after averaging their completion-token losses.

Table 8: Training populations for the one-pass matched comparison.

Population	Buckets	Examples per bucket	Total records
De novo specialist	6	81,415	488,490
Editing specialist	7	81,415	569,905
Joint / MOSA-Fixed / MOSA	13	81,415	1,058,395

Panel identity and isolation. Generation uses six groups of 1,000 requests. Editing uses ten tasks of 500 sources. Ordered request IDs, model/controller identities, scorer hashes and decoding settings are retained with the results. The available identity checks do not establish molecule- or scaffold-disjointness across training and evaluation, and base-model pretraining exposure is unknown. We make no such generalization claim.

Raw@1 decoding. For the main benchmarks, each request produces one saved completion, without retries, property reranking or verifier-guided repair. The separate budgeted inference study in Appendix G.2 uses candidate selection. Invalid and missing completions remain in the denominator. Batch seeds equal the base seed plus the mode offset and original batch-start index. Sharding preserves those boundaries and the ordered request IDs. With V_i denoting valid parsing, P_i joint

Table 9: Frozen sampled Raw@1 decoding contract.

Setting	Value
Temperature	0.8
Top-p / top-k	0.95 / 50
Repetition penalty	1.05
Maximum new tokens	128
Batch size	8
Beams / returned sequences	1 / 1
Base evaluation seed	36051
De novo / editing mode offset	0 / 100000
Padding side	Left

property success, and $R_i(\gamma)$ the source/output similarity test, scores are

$$A_{\text{gen}} = N^{-1} \sum_i V_i P_i, \qquad A_{\text{edit}}(\gamma) = N^{-1} \sum_i V_i P_i R_i(\gamma). \tag{4}$$

Editing uses radius-two, 2,048-bit Morgan Tanimoto similarity, with a strict threshold of 0.65 and an auxiliary relaxed threshold of 0.15. Macro and request-level means coincide within the balanced panels. Combining one-, two-, and three-property editing means uses weights 4:4:2.

Code availability. The anonymous implementation is available at `https://anonymous.4open.science/r/MOSA-60C6/`. It includes the method, training and evaluation code, main configurations, and tests. The release also identifies the MOSA-Fixed configuration used in the extended comparison. Datasets, evaluation panels, model weights, and third-party assay predictors are not bundled. The repository lists the required artifacts and its reproduction scope.

B Architecture and training details

Each projection stores shared, generation-private and editing-private rank-eight blocks. A request activates the shared block and one private block. LoRA scaling is $48/24 = 2$. The implementation computes 24 rank features before masking, with 16 active. LoRA contributes 71,384,064 trainable parameters and controllers 377,984, totaling 71,762,048.

Property features and symmetry. A 16-dimensional property embedding is concatenated with four goal channels: increase, decrease, numerical-goal indicator, and $\operatorname{asinh}(t/s_p)$. The condition encoder produces $\phi_i \in \mathbb{R}^{32}$. The interaction summary in the main method is

$$u = \frac{(\sum_i \phi_i)^{\odot 2} - \sum_i \phi_i^{\odot 2}}{\max\{k(k-1), 1\}} = \frac{\sum_{i \neq j} \phi_i \odot \phi_j}{k(k-1)} \quad (k > 1). \tag{5}$$

For a singleton, $u = 0$, although affine biases may still produce nonzero interaction outputs. Aggregation uses shared encoders and sums over active conditions. Permuting them therefore preserves the auxiliary mean, interaction summary and coefficients in exact arithmetic. This invariance concerns the controller. The backbone still reads the serialized token order. Shared coefficient coordinates lie

Table 10: Controller dimensions in the implemented model. Affine layers include biases.

Component	Dimensions / operation
Property embedding	8 entries $\times$ 16 (index 7 is padding/unknown)
Goal features	Increase, decrease, around, scaled target: 4 channels
Condition encoder	$20 \to 32$, GELU, LayerNorm
Interaction encoder	$32 \to 32$, GELU
Source encoder	$2048 \to 32$, GELU, LayerNorm
Source-condition encoder	Concatenated $64 \to 32$, GELU
Per-projection property head	$32 \to 8$
Per-projection interaction head	$32 \to 8$
Per-projection source head	$32 \to 16$, split into scale and shift

in $(0, 2.5)$. For each adapted projection ℓ, affine heads P_ℓ and I_ℓ and the interaction encoder ψ give

$$g_c^{(\ell)}(C) = \left[\frac{1}{k}\sum_i 2\sigma(P_\ell \phi_i)\right] \odot \left[1 + 0.25\tanh\left(I_\ell \psi(u)\right)\right]. \tag{6}$$

Source features and initialization. The source fingerprint is normalized by $\sqrt{\max(\sum_j f_j, 1)}$, mapped to 32 dimensions, and combined with the mean condition representation. The source head yields eight values each for a and b, with $g_e = 1 + 0.5\tanh(a)$ and $b_e = 0.1\tanh(b)$. Editing scales lie in $(0.5, 1.5)$ and rank-space shifts in $(-0.1, 0.1)$. The editing mask suppresses this branch for de novo requests. Zero-initialized controller heads yield unit scales and zero shifts. Zero LoRA up projections contribute nothing initially. The two private down projections start from the same private rank block and subsequently train by task.

Table 11: Registered MOSA training settings.

Setting	Value
Training seeds	36001 (MOSA repeats: 36002, 36003)
Effective batch	65 records
Maximum sequence length	448 tokens
Peak learning rate	2×10^{-5}
Weight decay	0.01
Gradient clipping norm	1.0
LoRA rank / alpha / dropout	24 / 48 / 0.05
Shared / each private rank	8 / 8
One-pass joint endpoint	16,283 updates
Separation coefficient	0.01 after warmup
Frozen backbone precision	BF16
Trainable adapter precision	FP32

Loss and implementation. Completion-token cross entropy is averaged within each record and then across records. Prompt and padding positions are masked. For row-normalized down projections $\widehat{A}$, the separation term is

$$\mathcal{L}_{\text{SEP}} = \text{mean}_{\ell,\, t\in\{d,e\}} \frac{\|\widehat{A}_c^{(\ell)}(\widehat{A}_t^{(\ell)})^\top\|_F^2}{64}. \tag{7}$$

Its coefficient warms linearly to 0.01 over 100 updates. Controller context is recomputed within each checkpointed LoRA hook during training. Evaluation-mode caching resets for every request batch.

Exposure and continuation. For an arm with b_a buckets, the update count at exposure q is $\lceil b_a q/65 \rceil$. Joint models have 13 buckets. Specialists have six or seven. This matches own-task records, not total updates, tokens or compute. The original schedule tracks consumed examples

Table 12: Exposure-matched checkpoints. The specialist counts at the final row include a short final batch.

q per bucket	Joint updates	De novo updates	Editing updates
16,250	3,250	1,500	1,750
32,500	6,500	3,000	3,500
48,750	9,750	4,500	5,250
65,000	13,000	6,000	7,000
81,415	16,283	7,516	8,768

divided by the arm's frozen row count, with warmup fraction $6500/1{,}058{,}395$ and linear decay. Continuation restores adapter/controller, optimizer, scheduler, Python/NumPy/Torch/CUDA RNG, global step and data position. Interrupted/uninterrupted checks compare consumed samples, learning rates, losses and saved states. Extended training has a separately registered schedule and selection rule.

C MOSA ALGORITHM

Algorithm 1 makes the request-dependent operations explicit. Its rank coordinates are LoRA features computed from a projection's hidden state; the transformer's hidden dimensions are not partitioned.

Algorithm 1 MOSA inference for one molecular request

Require: Property conditions $C = \{c_i\}_{i=1}^{k}$; optional source S; frozen transformer $\mathcal{T}$; trained LoRA branches and controllers
Ensure: One completion containing a plan label and target SMILES

$m \leftarrow \mathbf{1}[S \neq \varnothing]$; $x \leftarrow \mathrm{Serialize}(C, S)$
$\phi_i \leftarrow \mathrm{ConditionEncoder}(c_i)$ for each $c_i \in C$
$\bar{\phi} \leftarrow k^{-1} \sum_i \phi_i$; $u \leftarrow \big((\sum_i \phi_i)^{\odot 2} - \sum_i \phi_i^{\odot 2}\big) / \max\{k(k-1), 1\}$
if $m = 1$ **then**
    $v \leftarrow \mathrm{SourceConditionEncoder}(\bar{\phi}, \mathrm{SourceEncoder}(\mathrm{NormalizedMorgan}_{2,2048}(S)))$
end if
for each adapted projection ℓ **do**
    $g_c^\ell \leftarrow \mathrm{PropertyController}_\ell(\bar{\phi}, u)$
    if $m = 1$ **then**
        $(g_e^\ell, b_e^\ell) \leftarrow \mathrm{EditController}_\ell(v)$
    end if
    $\Delta_\ell(h) \leftarrow B_c^\ell(g_c^\ell \odot A_c^\ell h)$
    if $m = 0$ **then**
        $\Delta_\ell(h) \leftarrow \Delta_\ell(h) + B_d^\ell A_d^\ell h$
    else
        $\Delta_\ell(h) \leftarrow \Delta_\ell(h) + B_e^\ell(g_e^\ell \odot A_e^\ell h + b_e^\ell)$
    end if
    replace $W_\ell h$ by $W_\ell h + 2\Delta_\ell(h)$ in $\mathcal{T}$
end for
return one sampled plan label and SMILES from $\mathcal{T}(x; \{\Delta_\ell\})$

The controllers are evaluated once per request and reused across its tokens. Training uses the same projection updates under teacher forcing, with completion-token loss and the shared/private separation penalty (Section 3); only the LoRA branches and controllers are updated. Main evaluation takes one sampled completion without repair or reranking. Architecture dimensions and decoding settings appear in Appendices B and A.

D ONE-PASS CHECKPOINTS AND COMPLETE-PANEL RESULTS

The main external comparison uses MOSA at update 39,500. This section documents a different question: how MOSA, Joint and task specialists compare during the matched one-pass training schedule ending at update 16,283. Appendix E.1 separately compares the extended MOSA and MOSA-Fixed runs. Their checkpoint results do not form one interchangeable trajectory.

D.1 CHECKPOINT RESULTS AT MATCHED OWN-TASK EXPOSURE

Tables 13–17 report 2p–7p generation and ten editing tasks for MOSA, Joint and specialists. Joint and MOSA share update counts. Specialists match the number of examples seen from their own tasks. Every generation table uses the same 600 requests from the main test panel. Editing uses the complete 5,000-request panel at updates 3,250 and 16,283. At 6,500, 9,750 and 13,000 updates, only a 200-request editing probe is available. Those three editing columns show coarse intermediate estimates and should not be read as a continuous 5,000-request trajectory. Figure 5 uses a different, fixed development panel of 600 generation and 500 editing requests. Its generation rates are not directly interchangeable with the tables here.

Table 13: One-pass checkpoint at Joint/MOSA update 3,250: strict success (%) on 600 de novo requests and the 5,000-request editing panel.

De novo (600 requests; 100 per property count)

Method	2p	3p	4p	5p	6p	7p	Mean
Specialist	38.0	20.0	11.0	10.0	10.0	10.0	16.50
Joint	46.0	24.0	18.0	**16.0**	5.0	**19.0**	21.33
MOSA	**57.0**	**29.0**	**28.0**	15.0	**17.0**	13.0	**26.50**

Editing (5,000 requests; 500 per task)

Requested change	Specialist	Joint	MOSA
GSK3B up	**38.4**	32.6	35.6
RB down	59.4	56.0	**61.2**
MW up	50.0	48.6	**54.4**
SA down	57.4	52.8	**61.2**
HBA down, SA down	55.4	52.8	**56.2**
QED up, SA down	59.8	56.2	**63.4**
HBA down, LogP up	30.0	29.4	**33.4**
HBA down, MW down	58.8	56.2	**59.6**
DRD2 down, MW down, SA down	56.0	55.8	**59.6**
HBA up, MW up, QED down	46.0	40.4	**53.0**
Macro average	51.12	48.08	**53.76**

Table 14: One-pass checkpoint at Joint/MOSA update 6,500: strict success (%) on 600 de novo requests and the 200-request editing probe.

De novo (600 requests; 100 per property count)							
Method	2p	3p	4p	5p	6p	7p	Mean
Specialist	57.0	34.0	30.0	21.0	21.0	21.0	30.67
Joint	49.0	37.0	44.0	20.0	24.0	21.0	32.50
MOSA	**60.0**	**49.0**	**47.0**	**30.0**	**33.0**	**28.0**	**41.17**

Editing (200 requests; 20 per task)			
Requested change	Specialist	Joint	MOSA
GSK3B up	35.0	**45.0**	35.0
RB down	70.0	**90.0**	65.0
MW up	**55.0**	50.0	50.0
SA down	50.0	55.0	**60.0**
HBA down, SA down	60.0	65.0	**75.0**
QED up, SA down	**60.0**	55.0	55.0
HBA down, LogP up	25.0	35.0	**45.0**
HBA down, MW down	50.0	50.0	**55.0**
DRD2 down, MW down, SA down	**70.0**	60.0	**70.0**
HBA up, MW up, QED down	**60.0**	55.0	55.0
Macro average	53.50	56.00	**56.50**

Table 15: One-pass checkpoint at Joint/MOSA update 9,750: strict success (%) on 600 de novo requests and the 200-request editing probe.

De novo (600 requests; 100 per property count)							
Method	2p	3p	4p	5p	6p	7p	Mean
Specialist	**69.0**	40.0	44.0	26.0	31.0	28.0	39.67
Joint	58.0	45.0	43.0	34.0	35.0	33.0	41.33
MOSA	62.0	**50.0**	**57.0**	**41.0**	**41.0**	**36.0**	**47.83**

Editing (200 requests; 20 per task)			
Requested change	Specialist	Joint	MOSA
GSK3B up	40.0	**55.0**	35.0
RB down	**75.0**	65.0	70.0
MW up	70.0	45.0	**75.0**
SA down	**55.0**	50.0	50.0
HBA down, SA down	**70.0**	**70.0**	**70.0**
QED up, SA down	65.0	65.0	**75.0**
HBA down, LogP up	55.0	45.0	**60.0**
HBA down, MW down	**70.0**	65.0	65.0
DRD2 down, MW down, SA down	**70.0**	55.0	65.0
HBA up, MW up, QED down	45.0	**50.0**	45.0
Macro average	**61.50**	56.50	61.00

Table 16: One-pass checkpoint at Joint/MOSA update 13,000: strict success (%) on 600 de novo requests and the 200-request editing probe.

De novo (600 requests; 100 per property count) Method	2p	3p	4p	5p	6p	7p	Mean
Specialist	67.0	48.0	52.0	36.0	33.0	30.0	44.33
Joint	65.0	**57.0**	38.0	31.0	**45.0**	30.0	44.33
MOSA	**76.0**	54.0	**54.0**	**41.0**	43.0	**45.0**	**52.17**

Editing (200 requests; 20 per task) Requested change	Specialist	Joint	MOSA
GSK3B up	30.0	**40.0**	**40.0**
RB down	**80.0**	**80.0**	75.0
MW up	60.0	55.0	**65.0**
SA down	50.0	**75.0**	70.0
HBA down, SA down	75.0	70.0	**80.0**
QED up, SA down	**85.0**	65.0	75.0
HBA down, LogP up	50.0	35.0	**55.0**
HBA down, MW down	**70.0**	65.0	**70.0**
DRD2 down, MW down, SA down	65.0	50.0	**70.0**
HBA up, MW up, QED down	**70.0**	60.0	60.0
Macro average	63.50	59.50	**66.00**

Table 17: One-pass checkpoint at Joint/MOSA update 16,283: strict success (%) on 600 de novo requests and the 5,000-request editing panel.

De novo (600 requests; 100 per property count) Method	2p	3p	4p	5p	6p	7p	Mean
Specialist	66.0	52.0	55.0	28.0	34.0	**40.0**	45.83
Joint	65.0	52.0	48.0	**47.0**	47.0	39.0	49.67
MOSA	**74.0**	**59.0**	**62.0**	46.0	**48.0**	38.0	**54.50**

Editing (5,000 requests; 500 per task) Requested change	Specialist	Joint	MOSA
GSK3B up	46.0	46.2	**49.4**
RB down	77.8	73.0	**81.4**
MW up	67.0	63.8	**68.6**
SA down	71.6	61.2	**74.0**
HBA down, SA down	72.8	67.8	**75.4**
QED up, SA down	76.2	75.0	**81.4**
HBA down, LogP up	57.2	52.0	**58.0**
HBA down, MW down	75.0	68.6	**77.4**
DRD2 down, MW down, SA down	78.4	70.6	**79.2**
HBA up, MW up, QED down	67.8	66.8	**70.4**
Macro average	68.98	64.50	**71.52**

D.2 Complete one-pass endpoint

At update 16,283, de novo evaluation covers 6,000 requests, or 1,000 per property count. Editing covers ten tasks, with 500 source requests per task. All three methods use the same requests and Raw@1 scoring protocol. Strict editing success requires both the requested property changes and source similarity of at least 0.65. Main Table 2 gives strict success and transfer differences. Table 18 reports validity and relaxed editing success; Table 19 gives the full generation breakdown, while Table 17 already contains the full editing breakdown.

Paired uncertainty. Main Table 3 reports the complete-panel paired intervals. We use 2,000 cluster-bootstrap replicates with seed 36051, keeping paired model outcomes together. Generation clusters are request/condition IDs; editing clusters are canonical sources. Each replicate recomputes macro strict-rate differences. Replicates missing a required group denominator are excluded. The

Table 18: Additional metrics at the matched one-pass endpoint. All entries are percentages on the complete panels. Relaxed editing success ($A_{.15}$) requires validity, all property goals, and source similarity of at least 0.15. Main Table 2 reports strict success.

	De novo	Editing	
Model	Validity	Validity	$A_{.15}$
Specialist	97.0	96.6	86.1
Joint	97.0	96.6	85.1
MOSA	**97.1**	**97.2**	**86.9**

Table 19: Complete de novo strict success (%) at update 16,283: 6,000 requests, 1,000 per property count. The complete editing breakdown is in Table 17.

De novo (6,000 requests; 1,000 per property count)							
Method	2p	3p	4p	5p	6p	7p	Mean
Specialist	66.0	53.0	46.0	39.2	33.1	27.7	44.17
Joint	71.9	56.1	47.9	42.6	38.9	32.3	48.28
MOSA	**74.0**	**62.3**	**54.1**	**46.2**	**41.8**	**37.8**	**52.70**

pointwise 95% intervals use the 2.5th and 97.5th percentiles, without multiplicity correction. They describe panel variation conditional on the trained models.

D.3 Training-seed variation

Three MOSA runs reach update 16,283 under the same training and evaluation protocol. Table 20 gives their complete-panel results and sample standard deviations.

Table 20: Three independent MOSA training seeds at the one-pass update 16,283. Strict success (percent) uses identical 6,000-generation and 5,000-editing panels; SD is the sample standard deviation.

Training seed	De novo	Editing
36001	52.70	71.52
36002	54.45	70.78
36003	53.33	71.64
Mean $\pm$ SD	53.49 ± 0.89	71.31 ± 0.47

D.4 Visual summaries

Figure 5 shows matched own-task trajectories on the separate 600-generation/500-editing development panel. Figure 6 shows how generation changes over the five checkpoints on one fixed 600-request panel. Figure 7 shows the complete 5,000-request editing task breakdown. The tables above give the exact rates for all methods and panels.

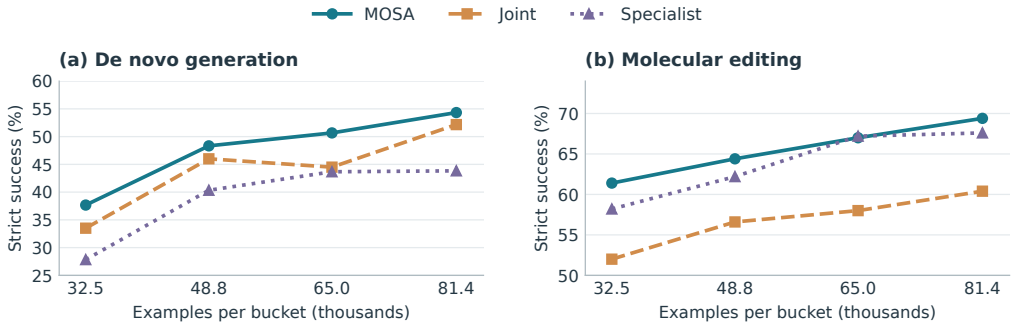


Figure 5: Matched own-task trajectories on a separate development panel, fixed across checkpoints: (a) 600 de novo and (b) 500 editing requests. The horizontal axis counts examples per property bucket. Specialists train on their own tasks. The Appendix checkpoint tables use a different 600-request generation subset of the main test panel.



Figure 6: One-pass de novo strict success by requested property count. Every point uses the same fixed 600-request panel, with 100 requests per property count. The horizontal axis gives task-matched examples per bucket; Joint and MOSA share the update index.

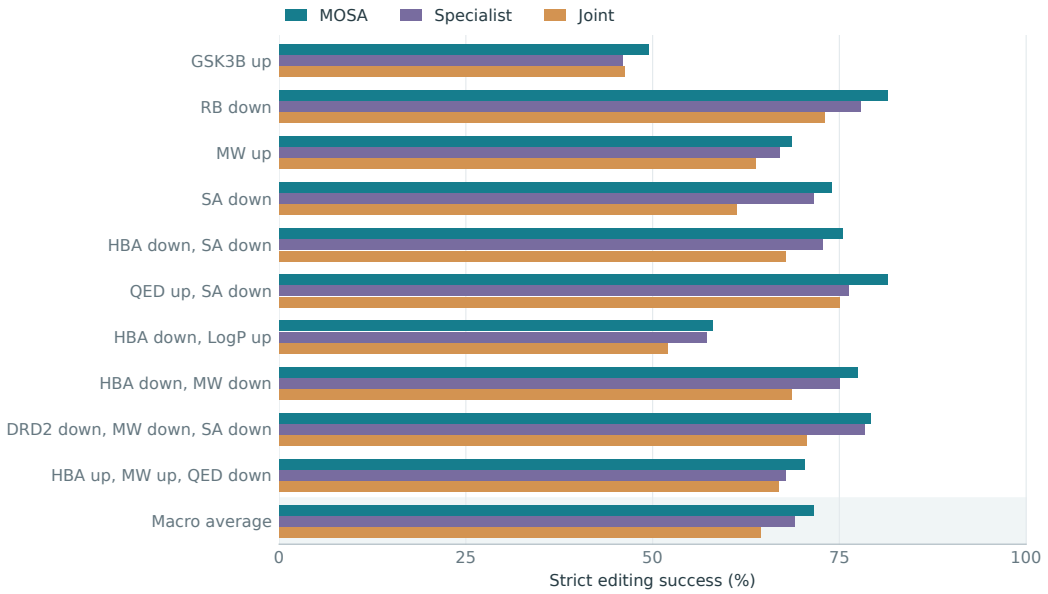


Figure 7: Complete one-pass editing endpoint by requested task. Each task has 500 requests on the common 5,000-request panel; the macro average weights the ten tasks equally.

E Design and retraining controls

E.1 Learned versus fixed composition

MOSA and MOSA-Fixed are trained independently on the same requests with the same frozen backbone, 8+8+8 shared/private LoRA layout and separation loss. MOSA-Fixed keeps auxiliary

scales at one and shifts at zero, while still reading the property program and source SMILES in its prompt. MOSA learns those coefficients and adds controller parameters and compute. This contrast tests the trained configurations, not coefficient variation at matched total capacity.

Table 21 follows both models on fixed 600-generation/500-editing development requests. That panel guided checkpoint selection. The complete panels provide the primary estimate. At 39,500 updates, MOSA scores 60.45% generation and 76.98% editing, compared with 59.40% and 76.22% for MOSA-Fixed on 6,000/5,000 requests. The generation difference is +1.05 points (95% paired interval $[-0.23, 2.30]$). Table 22 gives every property count and editing task. For context, at 16,283 updates the complete-panel differences are +0.37 generation and +0.78 editing points, with paired intervals $[-0.78, 1.48]$ and $[-0.11, 1.63]$.

Table 21: Equal-update comparison on identical 600-generation/500-editing development requests. Strict success is in percent, and Δ is MOSA minus MOSA-Fixed in points. This panel informed checkpoint selection. The complete-panel 39,500-step comparison is reported in Section 5.4.

	De novo (600)			Editing (500)		
Update	Fixed	MOSA	Δ	Fixed	MOSA	Δ
16,283	52.50	54.33	+1.83	69.40	69.40	0.00
28,496	56.67	58.00	+1.33	70.20	71.20	+1.00
39,500	57.33	63.17	+5.83	72.20	75.20	+3.00

Table 22: Equal-update complete-panel comparison at update 39,500. Strict success is in percent on identical 6,000-generation and 5,000-editing requests; Δ is MOSA minus MOSA-Fixed in percentage points. Best values within each comparable row are bolded, including ties.

De novo (1,000 requests per property count)			
Property count	MOSA-Fixed	MOSA	Δ (pp)
2 properties	**76.9**	76.2	-0.7
3 properties	67.5	**70.3**	+2.8
4 properties	63.2	**63.3**	+0.1
5 properties	**57.9**	56.8	-1.1
6 properties	46.4	**50.0**	+3.6
7 properties	44.5	**46.1**	+1.6
Macro average	59.40	**60.45**	+1.05
Editing (500 requests per task)			
Requested change	MOSA-Fixed	MOSA	Δ (pp)
GSK3B up	55.2	**55.6**	+0.4
RB down	**85.4**	**85.4**	+0.0
MW up	73.4	**75.4**	+2.0
SA down	**78.0**	**78.0**	+0.0
HBA down, SA down	78.2	**82.8**	+4.6
QED up, SA down	85.0	**85.2**	+0.2
HBA down, LogP up	**64.2**	63.6	-0.6
HBA down, MW down	**82.6**	82.4	-0.2
DRD2 down, MW down, SA down	85.0	**85.8**	+0.8
HBA up, MW up, QED down	75.2	**75.6**	+0.4
Macro average	76.22	**76.98**	+0.76

E.2 SOURCE MODULATION

Retrained control. The separately trained no-source-modulation arm reaches update 16,283 on the same 6,000/5,000-request panels. Source SMILES remain in the prompt; only editing-private scales and shifts are fixed to one and zero. Table 23 reports the complete paired comparison. Its editing difference favors MOSA by 1.46 points [0.58, 2.38], while the generation interval includes zero.

Table 23: Full-budget source-modulation retraining at update 16,283. Strict success uses the fixed main panels. Differences are MOSA minus ablation, with pointwise 95% paired cluster-bootstrap intervals. Source text is retained in both models.

Configuration	Generation (6,000)	Editing (5,000)
MOSA	52.70	71.52
Without source modulation	53.07	70.06
Difference (percentage points)	-0.37 [-1.57, 0.82]	$+1.46$ [0.58, 2.38]

Frozen-checkpoint controls. Here we change coefficients at evaluation time, without retraining. Identity coefficients reduce editing success from 71.52% to 61.72%. Calibrated mean and mismatched-source coefficients yield 71.08% and 71.24%; their paired intervals relative to MOSA include zero. These controls show that simply removing the learned coefficient pathway is disruptive. They do not show that variation across individual sources is required.

Table 24: Full-panel frozen source controls. Rates are percentages. Differences are intervention minus full MOSA, with pointwise paired 95% source-cluster intervals (percentage points).

Configuration	Validity	Property	Strict	Δ [95% CI]
Full MOSA	97.20	86.94	71.52	–
Identity coefficients	96.92	84.54	61.72	-9.80 [-11.08, -8.55]
Calibrated source mean	97.12	86.96	71.08	-0.44 [-0.89, 0.01]
Mismatched source	97.30	86.84	71.24	-0.28 [-0.73, 0.16]

E.3 Stored-adapter capacity

Joint-24 reaches update 16,283 with effective batch 65 and the same 71,384,064 stored LoRA parameters as MOSA. MOSA adds 377,984 controller parameters and activates rank 16; Joint-24 activates rank 24. The latter retains the original rank-16 down-projection rows, initializes eight additional rows independently and starts all up-projection rows at zero. Main Section 5.4 reports the scores and paired intervals. Equal LoRA storage does not imply equal total parameters or compute.

F Failure analysis and molecular examples

F.1 Editing outcomes and source retention

The editing counts below expand main Figure 3. Exclusive outcomes are assigned in this order: invalid, property failure, property pass with structural failure, then strict success. Every row includes all 5,000 requests.

Table 25: Exclusive editing outcomes at similarity 0.65 for one-pass endpoints, with 5,000 requests per arm. Failure columns are mutually exclusive.

Method	Strict success	Structure fail	Property fail	Invalid
MOSA	3576	771	513	140
MOSA-Fixed	3537	805	514	144
Joint	3225	1032	571	172
Specialist	3449	855	525	171

The threshold table rescores these same outputs while requiring every requested property at each threshold.

Table 26: Strict editing success (percent) on the complete threshold grid. Last column: MOSA minus Joint, percentage points and pointwise 95% interval.

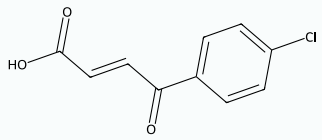
Threshold	MOSA	MOSA-Fixed	Joint	Specialist	Difference [95% CI]
0.00	86.94	86.84	85.14	86.08	+1.80 [0.75, 2.85]
0.20	86.78	86.68	84.86	86.00	+1.92 [0.85, 3.00]
0.30	86.38	86.32	84.08	85.72	+2.30 [1.23, 3.40]
0.40	85.52	85.34	82.42	84.70	+3.10 [2.00, 4.23]
0.50	83.32	83.10	79.18	82.04	+4.14 [3.01, 5.36]
0.60	78.24	77.36	71.72	76.16	+6.52 [5.22, 7.73]
0.70	42.42	42.22	39.24	41.24	+3.18 [2.12, 4.31]
0.80	3.20	3.16	3.48	3.32	-0.28 [-0.67, 0.11]
0.90	0.02	0.00	0.08	0.02	-0.06 [-0.16, 0.02]

F.2 MOLECULAR EXAMPLES

Figures 8 and 9 show saved Raw@1 outputs, without candidate reranking. Generation cases are the first eligible requests in the indicated success or valid-property-failure groups; the editing cases follow the saved case manifest's fixed selection. The manifest retains request IDs, source and output SMILES, checkpoint identities and prediction hashes. Displayed descriptor values are rounded, but scoring uses unrounded values. Cases illustrate outcomes, not their prevalence.

2 properties / strict success

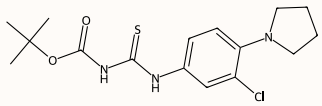
denovo_2p_000000



Property	Target	Output	Result
LogP	1.69	2.16	pass
RB	3	3	pass

2 properties / valid property failure

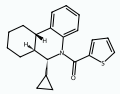
denovo_2p_000002



Property	Target	Output	Result
QED	0.653	0.780	fail
HBA	4	4	pass

7 properties / strict success

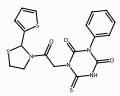
denovo_7p_000000



Property	Target	Output	Result
MW	334.55	337.49	pass
LogP	4.95	5.46	pass
QED	0.734	0.712	pass
TPSA	20.23	20.31	pass
HBD	1	0	pass
HBA	3	2	pass
RB	1	2	pass

7 properties / valid property failure

denovo_7p_000001



Property	Target	Output	Result
MW	429.48	432.55	pass
LogP	2.32	2.39	pass
QED	0.538	0.641	fail
TPSA	83.08	80.10	pass
HBD	1	1	pass
HBA	7	8	pass
RB	4	4	pass

Figure 8: Saved two- and seven-property MOSA generations. Each pair places a strict success beside a valid output that misses a requested condition. Targets, recomputed values and individual verdicts are shown.

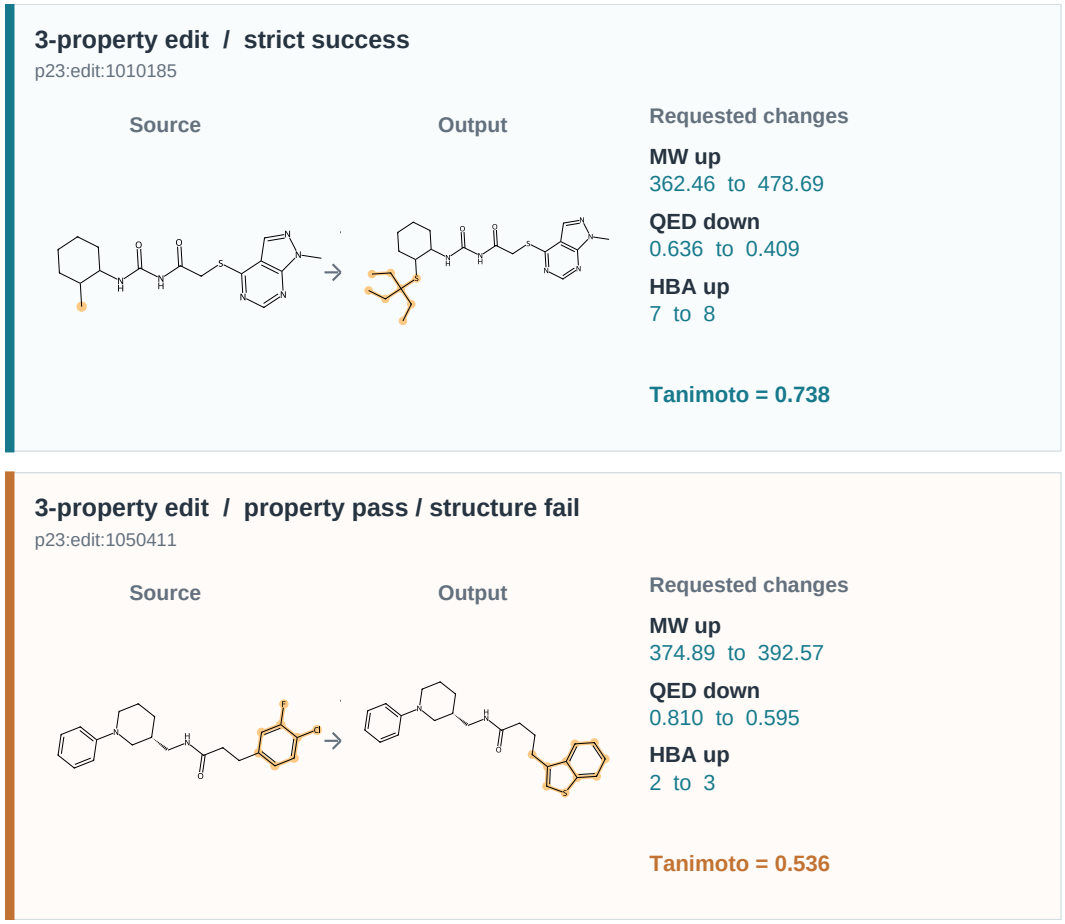


Figure 9: Saved three-property MOSA edits. Both outputs satisfy the requested directions, but the lower example fails the 0.65 source-similarity threshold. Orange atoms fall outside an exact-atom maximum common substructure; highlighting does not encode fingerprint similarity.

G Additional evaluations

G.1 External reference protocols

External generation uses 6,000 common requests and one output per request. Methods retain their native training data, representation and decoding. The released SPMM checkpoint masks unspecified property-vector entries and uses its native stochastic decoder ($k = 2$) without reranking; it reaches 45.62% strict success and 97.57% validity. The available Graph DiT epoch-100 checkpoint reaches 7.28% strict success and 28.82% validity under its native graph output pipeline. This is an early checkpoint in its 10,000-epoch training schedule. Its graph representation omits charge and stereochemistry. Ordered panel IDs, checkpoint identities and scorer hashes are retained with both results.

MolEditRL editing values in Table 27 are published public references. MOSA uses the declared 5,000-request panel and its 39,500-update checkpoint. The side-by-side values are descriptive; paired intervals are used only for internal comparisons on identical requests. The ten task-wise values are preserved in the table and its source data.

Table 27: Task-wise strict editing success (percent). MOSA is evaluated at update 39,500, with 500 requests per task. MolEditRL scores are public references.

ID	Requested change	MOSA (extended)	MolEditRL
E1	GSK3B up	55.6	34.2
E2	RB down	85.4	63.4
E3	MW up	75.4	40.4
E4	SA down	78.0	62.8
E5	HBA down, SA down	82.8	34.6
E6	QED up, SA down	85.2	63.2
E7	HBA down, LogP up	63.6	31.6
E8	HBA down, MW down	82.4	25.2
E9	DRD2 down, MW down, SA down	85.8	51.8
E10	HBA up, MW up, QED down	75.6	43.0

G.2 Budgeted sampling and iterative editing

This exploratory 500-source panel is separate from the main editing benchmark. Every source has a held-out reference satisfying all seven conditions and source similarity at least 0.65; references establish feasibility but are not used to generate or select candidates. The sources and references are absent from the full training stream. All arms use MOSA at update 16,283 and share their first candidate. Independent sampling draws again from the original source. Full-program iteration repeats all clauses from the best retained candidate; remaining-goal iteration requests only unmet clauses. Both iterative arms retain the original source when no eligible candidate is found.

Each arm receives the same one-, two-, or three-candidate budget. Selection first requires validity and similarity at least 0.65 to the *original* source, then ranks property satisfaction and breaks ties using mean satisfaction, similarity and candidate order. Final scores use the original source and all seven clauses. These are descriptor-guided selection results, not Raw@1 or reinforcement learning.

Table 28: Seven-property strict editing success (%) on the same 500 known-feasible sources. Budgets two and three include descriptor-guided candidate selection. All arms share the budget-one output.

Strategy	1 candidate	2 candidates	3 candidates
Independent sampling	53.4	65.8	72.6
Full-program iteration	53.4	64.2	70.2
Remaining-goal iteration	53.4	62.2	67.6

At three candidates, independent sampling reaches 72.6% strict success, full-program iteration 70.2% and remaining-goal iteration 67.6%. Their differences from independent sampling are -2.4 points (95% source-paired interval $[-5.6, 0.8]$) and -5.0 points ($[-8.0, -2.0]$). These heuristics show no multi-step advantage at equal candidate budget on this previously examined feasible panel. The 5,000-replicate intervals are pointwise and exploratory.

H Limitations and Future Directions

Limitations. The current study has several limitations:

- **Structured molecular instructions.** The current framework uses structured property programs rather than free-form instructions. Editing objectives specify directional changes, such as increasing or decreasing a property, and do not cover the full range of molecular design requests that may arise in practice.
- **Source representation and structural preservation.** Molecular editing currently represents the source through its SMILES sequence and a Morgan fingerprint. Source preservation is evaluated using fingerprint-based Tanimoto similarity, which does not fully capture scaffold identity, stereochemistry, or three-dimensional molecular structure.

- **Generalization.** The current evaluation does not establish molecule- or scaffold-disjoint generalization between training and evaluation data. The pretraining exposure of the frozen backbone is also unknown. We therefore do not make a strong out-of-distribution generalization claim.
- **Understanding negative transfer.** Our matched experiments identify negative transfer at the task-performance level, but they do not establish the optimization mechanism that produces it. In particular, the current study does not directly measure gradient conflict or how interference develops within the shared and private subspaces.
- **Computational molecular objectives.** Property descriptors, structural similarity, and activity predictors provide useful computational objectives, but they do not establish that generated molecules are synthesizable, safe, stable, or biologically effective.

Future directions. Several directions could extend the current framework:

- **More flexible molecular instructions.** Future work could move beyond structured property programs to support free-form molecular instructions, exact target values, and more diverse combinations of quantitative and qualitative design constraints.
- **Richer source representations.** Molecular editing could incorporate additional source information beyond SMILES and fingerprints. Molecular images could provide complementary structural information, and more general multimodal representations could combine sequence, graph, image, and three-dimensional information for source-conditioned adaptation.
- **Understanding negative transfer.** A deeper analysis could examine gradient interactions, representation changes, and the geometry of the learned shared and private subspaces. Such analysis may help explain when joint molecular training leads to positive transfer and when task interference emerges.
- **More capable multi-step editing.** The current results suggest that simple iterative editing does not automatically improve performance under an equal candidate budget. Future work could explore learned planning, verifier-guided refinement, reinforcement learning, or search-based strategies that decide how and when to perform successive molecular edits.