FORGE: REACTION-GUIDED GENERATIVE DESIGN OF IONIZABLE LIPIDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Ionizable lipids are essential for RNA delivery, but their discovery largely depends on synthesizing and screening libraries assembled from predefined reactants. This strategy preserves a defined assembly reaction while limiting the chemical space that can be explored. *De novo* molecular generation could expand this space, yet existing whole-molecule models do not retain the reaction information needed to assemble their designs. We introduce FORGE (**F**low-matched, **O**pen-ended, **R**eaction-guided **G**eneration and **E**xploration), a generative model that uses known reaction chemistry to design ionizable lipids without restricting generation to a fixed reactant catalogue. FORGE represents each assembly program through precursor roles, reaction-core positions, transformation order, and coarse molecular architecture, while generating the complete molecule at atom-and-bond resolution. The model does not receive component identifiers, stored precursor graphs, or fragment tokens. We evaluated FORGE on Ugi, repeated aza-Michael addition, and repeated reductive amination. Across three independent seeds, FORGE achieved exact level-one (exact-L1) transform-consistent yields of $96.4 \pm 2.1\%$, $72.2 \pm 1.6\%$, and $52.3 \pm 3.4\%$, respectively. Matched whole-molecule generation followed by *post hoc* reaction filtering yielded $34.8 \pm 7.1\%$, $19.2 \pm 4.1\%$, and $37.0 \pm 2.2\%$. FORGE also generated exact-L1 products with precursor identities absent from the corresponding training component catalogues across all three reactions, a capability unavailable to finite-catalogue assemblers. In an *in vivo* reporter mRNA-LNP study, three synthesized FORGE lipids demonstrated higher potency than the clinical benchmark MC3, with one lipid showing ~13-fold higher signal. These results establish that known reaction chemistry can guide *de novo* molecular generation while preserving the information needed to verify assembly and demonstrate that generated lipids can achieve strong *in vivo* performance.

1 INTRODUCTION

Lipid nanoparticles (LNPs) deliver nucleic-acid therapeutics including small interfering RNA, mRNA vaccines, and *in vivo* genome-editing reagents (Adams et al., 2018; Polack et al., 2020; Baden et al., 2021; Gillmore et al., 2021). Within each LNP, the ionizable lipid supports nucleic-acid complexation, particle assembly, and endosomal escape, making its chemical structure a major determinant of delivery (Cullis & Hope, 2017; Hou et al., 2021). Small changes to the headgroup or hydrophobic domains can substantially alter function. Their effects depend on the complete molecular structure and do not follow simple design rules (Cheng et al., 2020; Han et al., 2021; Hatit et al., 2023). Researchers therefore discover ionizable lipids by defining modular precursor libraries, synthesizing combinations, and screening the resulting LNPs experimentally (Akinc et al., 2008; Love et al., 2010; Li et al., 2024; Xu et al., 2024). This strategy preserves synthetic tractability and supports systematic testing within defined chemical families, while limiting the search to components selected before design begins.

De novo molecular generation has the potential to extend ionizable-lipid discovery beyond fixed precursor libraries. Several synthesis-aware generators construct molecules by selecting reactants and applying predefined transformations (Gao et al., 2022; Koziarski et al., 2024; Cretu et al., 2025). This construction retains an explicit assembly route and limits generated molecules to the supplied reactant inventory. SynCoGen instead jointly generates building-block graphs and reaction assignments (Rekesh et al., 2026). Whole-molecule generators can propose structures beyond a fixed inventory,

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.

but their outputs do not retain the precursor roles and transformations needed to assemble each design (Qin et al., 2025; Lee et al., 2025). Other approaches estimate synthesizability after generation without incorporating the intended assembly chemistry into the generative process (Gao & Coley, 2020). Recent studies have also applied *de novo* generative models to ionizable lipids (Ou et al., 2024; Maganti et al., 2026). Here we study generation of complete molecular structures under known reaction chemistry while leaving precursor identities open.

To mitigate this, we introduce FORGE (**F**low-matched, **O**pen-ended, **R**eaction-guided **G**eneration and **E**xploration), a generative model that uses known reaction chemistry to design ionizable lipids beyond a fixed reactant catalogue. Each reaction program contains the assembly chemistry, precursor roles, reaction-core positions, transformation order, and coarse molecular architecture while leaving precursor identities unspecified. FORGE generates complete molecular graphs at atom-and-bond resolution without component identifiers, stored precursor graphs, or fragment tokens. We then decompose each completed graph into its implied precursors and require exact forward replay of the specified assembly reaction to reconstruct the generated molecule.

We evaluate one shared FORGE model across a Ugi three-component reaction, repeated aza-Michael addition and repeated reductive amination. Across three independent seeds, verified exact-L1 yields were $96.4 \pm 2.1\%$, $72.2 \pm 1.6\%$ and $52.3 \pm 3.4\%$, respectively, exceeding matched whole-molecule generation followed by *post hoc* verification in each program. The generated assemblies also contained precursor identities outside the corresponding training catalogues. These experiments concern specified reaction programs within bounded graph support; they do not establish zero-shot support for arbitrary chemistry or strong recovery of designated held-out precursors. We additionally report single 4-hour FLuc imaging values for three synthesized FORGE lipids and a DLIN-MC3-DMA formulation, without inferring variability or biological potency from that record.

We summarize our main contributions as follows.

- FORGE generates complete molecular graphs using known reaction chemistry while leaving precursor identities open and does not select from stored reactants or fragment vocabularies.
- Matched controls across three assembly programs showed that reaction-program conditioning substantially increased exact transform-consistent generation relative to whole-molecule generation followed by *post hoc* filtering.
- Across all three programs, FORGE generated exact assemblies with recovered precursor identities outside the corresponding training component catalogues, extending verified component identities beyond finite catalogues under the evaluated protocols.
- Three generated lipids were synthesized and evaluated in a mouse FLuc reporter study, with the third exhibiting a ~13-fold increase in potency relative to the clinical benchmark lipid MC3.

2 Preliminaries and problem formulation

2.1 Ionizable lipids and molecular architecture

Ionizable lipids combine protonatable amine-containing headgroups, linkers and hydrophobic tails (Semple et al., 2010; Jayaraman et al., 2012). Protonation supports nucleic-acid association during acidic formulation and membrane interactions involved in endosomal release (Xu et al., 2024). LNPs commonly combine an ionizable lipid with helper lipids, cholesterol and PEGylated lipids. Lipid structure, stereochemistry and formulation composition affect delivery (Han et al., 2021; Hatit et al., 2023; Cheng et al., 2020); molecular constitution and formulated-particle performance are therefore distinct evaluation levels.

2.2 Reaction families and modular assembly

We study three modular assembly patterns. Ugi 3-CR combines an amine, aldehyde and isocyanide into an α-amino amide (Xu et al., 2024). Repeated aza-Michael addition attaches acrylate-bearing components to amine sites (Li et al., 2023); repeated reductive amination couples aldehyde-bearing components to amines through condensation and reduction (Xue et al., 2024). The Ugi variant has no

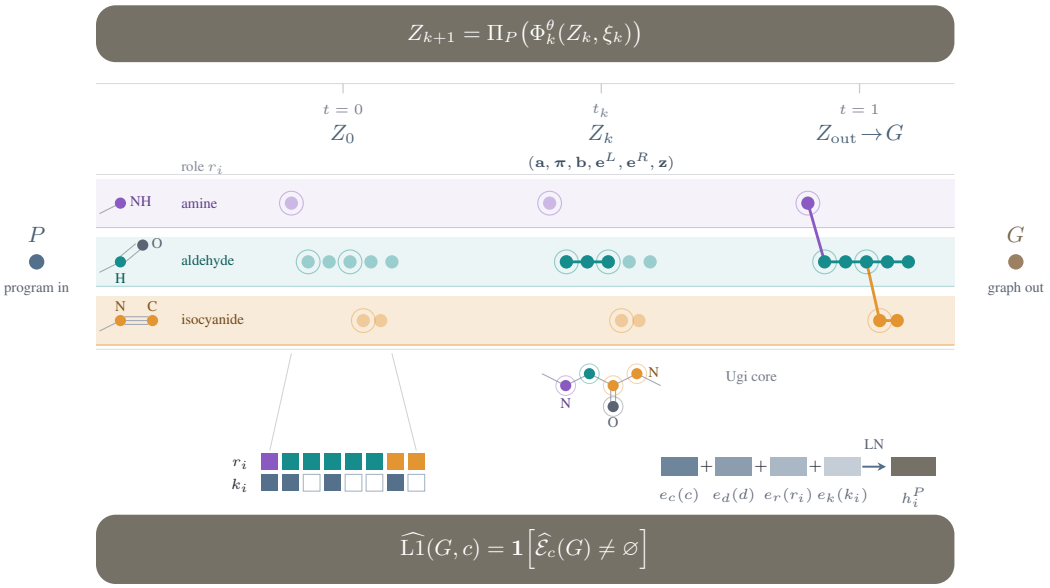


Figure 1: **FORGE generation and exact assembly verification.** A reaction program fixes assembly roles and core coordinates without selecting precursor identities. FORGE generates the whole graph, recovers its precursors and verifies that forward replay reconstructs the product. Colours mark precursor origin; grey denotes the Ugi assembly-introduced oxygen. Exact replay establishes computational assembly consistency.

carboxylic-acid reactant component and uses an acidic phosphorus catalyst; its ester functionality comes from the aldehyde precursor. Repeated assemblies require both transformation order and participating sites. Precursor-origin boundaries can span functional headgroup, linker and tail regions. These distinctions motivate role-assigned reaction programs (Appendix D.1, Table 12).

2.3 Molecular graphs and reaction programs

A lipid constitution is a nonempty, connected, simple molecular graph $G = (V, E, \ell_V, \ell_E)$, with atom labels $\ell_V : V \to \Sigma_{\text{atom}}$ and bond labels $\ell_E : E \to \Sigma_{\text{bond}}$ in finite vocabularies. Bond order is an edge label. We compare graphs up to label-preserving isomorphism after removing atom-map and stereochemical labels, retaining formal charges and bond orders. We write G for a graph or its constitutional isomorphism class when the distinction is immaterial. The bounded graph space $\mathfrak{G}_{N,K_{\text{cl}}}$ contains these classes with $1 \leq |V| \leq N$ and cycle rank $\beta(G) = |E| - |V| + 1 \leq K_{\text{cl}}$. This is a structural support definition; molecular validity additionally requires valence and sanitization checks. Here $N = N_{\max} = 194$ and $K_{\text{cl}} = 3$.

A reaction program specifies the assembly constraints,

$$P = (c, d, \mathbf{r}, \mathbf{k}, M). \tag{1}$$

Here c identifies a registered assembly chemistry with its admitted transformation sequences, d is the requested depth, $\mathbf{r}$ assigns precursor roles to atom positions, and $\mathbf{k}$ marks reaction-core positions. The four morphology fields in M record each role's exterior atom count, junction budget, closure count and core-attachment count. These quantities constrain coarse architecture without specifying precursor identities. Precursor origin and core membership are separate: an atom can belong to a precursor and participate in the core; a template-introduced atom belongs to the core without belonging to a precursor. Appendix A.1 defines the morphology counts.

2.4 Exact assembly and component novelty

For registered chemistry c, a precursor input $\mathbf{C} \in \mathfrak{C}_c$ is an ordered tuple of constitutions, roles and transformation steps. Its registered forward products within the bounded support form $\mathcal{F}_c(\mathbf{C}) \subseteq$

$\mathfrak{G}_{N,K_{\mathrm{cl}}}$. Reverse search returns a finite set $\mathcal{D}_c(G)$ of precursor inputs and assembly traces $(\mathbf{C}, \tau)$. Exact forward replay retains

$$\widehat{\mathcal{E}}_c(G) = \{(\mathbf{C}, \tau) \in \mathcal{D}_c(G) : G \in \mathcal{F}_c(\mathbf{C})\}. \tag{2}$$

Membership uses constitutional identity. A valid graph passes verified exact level-one assembly (L1) when $\widehat{\mathcal{E}}_c(G) \neq \varnothing$. Bounded reverse search can miss assemblies, and recovered traces need not match every sampled depth, role-position or morphology field in P. Assembly existence and conditioning adherence are distinct checks, as are decomposition coverage and replay precision (Appendix A.5).

A recovered precursor is component-novel when its constitution is absent from its role's catalogue. Strong product-level catalogue escape requires *every* exact precursor input to contain an outside component. We measure recovered component novelty, which does not establish that universal claim (Appendix C.2). The three-family study evaluates final-assembly consistency and component novelty. It does not evaluate preparation routes or supplier availability for the recovered precursors; exact replay alone does not establish experimental synthesis success or delivery activity.

3 FORGE

3.1 Overview

FORGE generates complete ionizable-lipid graphs under specified assembly chemistry while leaving precursor identities open. Let $G \in \mathcal{G}$ be a complete molecular graph and $P \in \mathcal{P}$ a reaction program. FORGE models $p_\theta(G \mid P)$ without receiving precursor identifiers, stored component graphs or fragment tokens. The program specifies how the molecule is assembled without selecting the precursor structures. Figure 1 summarizes three stages:

1. **Specify the reaction program.** Choose assembly chemistry, roles, reaction-core positions, transformation order and morphology, leaving precursor identities open.
2. **Generate the whole graph.** A conditional discrete flow generates atom, bond and topology states while preserving program-fixed coordinates.
3. **Verify exact assembly.** Recover role-assigned precursors and require their forward transform to reconstruct the generated molecule.

3.2 Conditioning on reaction programs

An admissible complete program is sampled from the training-fold prior,

$$p_\theta(G, P \mid c) = p_{\mathrm{prog}}(P \mid c)\, p_\theta(G \mid P). \tag{3}$$

Learned embeddings combine chemistry, depth, precursor role, core status and role-local morphology at each graph position. The adapter determines active positions and fixed or variable coordinate masks. Appendix A.1 specifies the embeddings and morphology counts.

3.3 Graph representation and discrete flow

FORGE represents each molecule as a spanning tree with additional bonds for ring closures. A deterministic encoder maps the molecular graph G and its selected atom-mapped assembly annotation α to

$$X_\star := \mathrm{Enc}_P(G, \alpha) = (\mathbf{a}, \boldsymbol{\pi}, \mathbf{b}, \mathbf{e}^L, \mathbf{e}^R, \mathbf{z}), \tag{4}$$

where $\mathbf{a}$ records atom states, $\boldsymbol{\pi}$ identifies each atom's parent in the tree, and $\mathbf{b}$ records the tree bonds. The arrays $\mathbf{e}^L$ and $\mathbf{e}^R$ identify the endpoints of additional bonds, whose states are stored in $\mathbf{z}$.

The padded state space $\overline{\mathfrak{X}}_{N,K_{\mathrm{cl}}}$ is finite. Its valid encodings form $\mathfrak{X}_{N,K_{\mathrm{cl}}}$: each contains a connected tree and at most K_{cl} distinct closure edges. The decoder Dec reconstructs the graph from a valid encoding and returns $\perp_{\mathrm{dec}}$ for a malformed state. Every graph within the specified atom, bond, size and closure limits has an encoding (Theorem C.3).

During sampling, we also restrict the allowed parents and number of children for each atom. These constraints limit which valid encodings the sampler can use.

For a given P, let $\mathcal{I}(P)$ index the categorical coordinates, each with finite alphabet $\mathcal{X}_i(P)$. Pointer alphabets contain only positions permitted by the layout. The adapter partitions coordinates into fixed positions $\mathcal{F}(P)$ with values $x_{P,i}$ and variable positions $\mathcal{V}(P)$. The set Ω_P contains raw states within the allowed alphabets that satisfy the fixed values. It can contain malformed or chemically invalid states. The source law q_0^P fixes $\mathcal{F}(P)$ and uses independent, full-support marginals $q_{0,i}^P$ on variable alphabets. For a clean endpoint $x_1 \in \Omega_P$, the conditional corruption law is

$$q_{t|1}^P(x \mid x_1) = \mathbf{1}[x_{\mathcal{F}(P)} = x_{P,\mathcal{F}(P)}] \prod_{i \in \mathcal{V}(P)} \left[(1-t)q_{0,i}^P(x_i) + t\mathbf{1}\{x_i = x_{1,i}\}\right], \quad 0 \le t \le 1. \tag{5}$$

Each coordinate interpolates between its source and clean value. Their conditional joint law is a product, so it generally differs from a linear mixture of the joint source and endpoint laws.

The clean-coordinate predictor $d_{\theta,i}(a \mid x, t, P)$ assigns a probability to each $a \in \mathcal{X}_i(P)$. A continuous-time Markov chain (CTMC) uses these predictions to average conditional one-coordinate rates,

$$R_{t,i}^\theta(u, v \mid x, P) = \sum_{a \in \mathcal{X}_i(P)} d_{\theta,i}(a \mid x, t, P) R_{t,i}^*(u, v \mid a, P), \qquad u \ne v, \quad t < 1. \tag{6}$$

The conditional rate $R_{t,i}^*$ realizes the corresponding factor in Equation 5; Appendix C.3 defines it and proves the path identity. The joint generator permits one coordinate to change at an infinitesimal time. Coordinate posterior marginals therefore suffice to specify its rates, even when endpoint coordinates are statistically dependent.

3.4 Program-conditioned Transformer and training

A graph Transformer predicts clean coordinates from (X_t, t, P). Each layer combines graph self-attention with parent, child and closure relation biases and cross-attention to program, depth, role and core tokens. Softly program-routed residual adapters add family-specific parameters. Qualified atom-mapped traces supervise role and core annotations; unresolved source-trace ambiguity excludes a record from semantic training.

The source-balanced training law μ_{bal} covers annotated triples (G, α, P) with equal mass across the three chemistries, not across individual morphology-specific programs. The final model combines masked coordinate cross-entropy, role-balanced chemistry, role/core supervision, and repeat and topology consistency terms. We combined chemistry-task gradients with PCGrad (Yu et al., 2020). Appendix A.3 specifies the full objective and training-time distribution.

To isolate the statistical property of coordinate denoising, define the reference population loss

$$\mathcal{L}_{\mathrm{ref}}(\theta) = \mathbb{E}_{\substack{(G,\alpha,P)\sim\mu_{\mathrm{bal}},\ t\sim\mathcal{U}(0,1) \\ X_t \sim q_{t|1}^P(\cdot \mid X_\star)}} \left[\sum_{i \in \mathcal{V}(P)} w_i(P) \left[-\log d_{\theta,i}(X_{\star,i} \mid X_t, t, P) \right] \right], \qquad w_i(P) > 0. \tag{7}$$

The weights depend only on the program; they can normalize coordinate counts without depending on the unknown clean category. This reference loss supports the population analysis. It is not the complete production training objective.

The reference objective has an ideal population interpretation. For a program with positive training mass, let q_1^P be the law of $\mathrm{Enc}_P(G, \alpha)$ under $\mu_{\mathrm{bal}}(\cdot \mid P)$, and let μ_G^P be the corresponding graph law.

Theorem 3.1 (Ideal conditional generation). *Fix such a program P, with registered chemistry c, finite coordinate alphabets and a full-support source on each variable alphabet. Assume every training endpoint belongs to Ω_P, decodes to its source graph, and passes molecular validity and verified exact L1. Assume an unrestricted measurable coordinate predictor attains a population minimum of Equation 7. Starting from q_0^P, exactly simulate the local CTMC of Equation 6 on $0 \le t < 1$. Then its marginal law converges to q_1^P as $t \uparrow 1$, and*

$$\mathrm{Dec}_\# q_1^P = \mu_G^P, \qquad \Pr_{G \sim \mu_G^P}\left[\mathrm{Valid}(G) = 1, \widehat{\mathrm{L1}}(G, c) = 1\right] = 1. \tag{8}$$

Here $\mathrm{Dec}_\#$ denotes the distribution obtained by decoding.

Appendix C.3 proves this result by recovering the clean-coordinate posteriors and verifying the CTMC forward equation. The theorem concerns the reference population objective and the limit of exact continuous-time sampling. It establishes neither generalization beyond q_1^P nor correctness of finite training, PCGrad, capped Euler updates or constrained terminal readout.

3.5 Sampling and exact assembly verification

The numerical sampler uses states Z_k, distinct from the ideal CTMC X_t. Starting from $Z_0 \sim q_0^P$, it applies

$$Z_{k+1} = \Pi_P\big(\Phi_k^\theta(Z_k, \xi_k)\big), \qquad k = 0, \dots, L_{\text{samp}} - 1, \tag{9}$$

where Φ_k^θ is a capped independent-coordinate Euler update, ξ_k contains its random draws and Π_P restores program-fixed coordinates. Fixed coordinates have zero rate and remain constant (Proposition C.6); chemical consistency requires verification. A fresh prediction at $t = 1$ drives the constrained argmax readout $Z_{\text{out}} = H_P^\theta(Z_{L_{\text{samp}}})$, including Ugi core-saturation rules, or a typed failure. We then decode, sanitize and apply Equation 2 to each valid graph. Molecular validation, decomposition, exact replay and root ambiguity are recorded separately. Appendix A.4 specifies the numerical operations and their distinction from the ideal CTMC.

4 Results

4.1 Evaluation protocol and matched budgets

We compared correctly conditioned, shared-null and cyclically mislabelled models using the same architecture, optimization budget, attempt count and final-checkpoint rule across three independent training seeds. Component-disjoint held-out products were excluded from training and checkpoint selection. Verified exact-L1 yield counts attempts passing molecular validation, decomposition and exact forward replay, with *all* attempts in the denominator. Post-hoc rejection cannot increase the proposal's per-attempt acceptance probability (Proposition C.10). We report means and sample standard deviations across training seeds, distinguishing retraining variation from conditional Monte Carlo variation. Appendix B.1 specifies the comparisons, coverage and precision metrics, and matched budgets.

4.2 Shared generation achieved exact assembly across three programs

We first tested whether one conditional flow could learn all three reactions while preserving Ugi performance. Across 3,072 held-out attempts per program and seed, FORGE achieved verified exact-L1 yields of $96.4 \pm 2.1\%$ for Ugi, $72.2 \pm 1.6\%$ for repeated aza-Michael addition and $52.3 \pm 3.4\%$ for repeated reductive amination (Table 1). Every verifier-returned trace passed exact forward replay in every program and seed, as required by construction. Reductive amination showed greater between-seed variation and higher decomposition ambiguity, indicating less consistent learning in this lower-data setting.

FORGE produced no fixed-coordinate violations or support overflows across 27,648 held-out attempts. These runs established exact assembly across all three programs, with the highest yield for Ugi.

Table 1: **Exact assembly under three ionizable-lipid reaction programs.** Exact-L1 yields are mean $\pm$ sample standard deviation across three seeds. The final column reports decomposition coverage and the forward-replay verifier invariant. Per-seed values: Appendix B.

Program	FORGE conditioned	Shared null	Cyclic program	Decomp. coverage / replay invariant (%)
Ugi	96.4 ± 2.1	34.8 ± 7.1	93.8 ± 4.2	99.9/100.0
Aza-Michael	72.2 ± 1.6	19.2 ± 4.1	71.5 ± 2.2	100.0/100.0
Reductive amination	52.3 ± 3.4	37.0 ± 2.2	50.7 ± 4.9	86.6/100.0

4.3 PROGRAM CONDITIONING IMPROVED EXACT-ASSEMBLY YIELD

To isolate the effect of reaction information during generation, we compared the conditioned model with the shared-null and cyclic arms under the same attempt budget and post-hoc verifier. Shared generation without program information yielded $34.8 \pm 7.1\%$, $19.2 \pm 4.1\%$ and $37.0 \pm 2.2\%$ exact L1 for Ugi, aza-Michael addition and reductive amination, respectively (Table 1). For Ugi, conditioning increased yield by 61.6 percentage points (three-seed range 57.4 to 67.7 points).

Assigning the wrong reaction identity reduced exact-L1 yield to $93.8 \pm 4.2\%$, $71.5 \pm 2.2\%$ and $50.7 \pm 4.9\%$ across the three programs. The cyclic control retained precursor roles, reaction-core coordinates and program depth, isolating the contribution of the reaction-identity label within a structured program. The larger conditioned-versus-null differences supported a benefit from program information during generation. The small conditioned-versus-cyclic differences did not establish a strong independent contribution from the identity label.

4.4 DECODER AND ARCHITECTURE EFFECTS DIFFERED ACROSS PROGRAMS

With seed-0 weights fixed, enforcing Ugi core hydrogen states increased exact-L1 yield with nearly unchanged molecular validity; changing source marginals from program-role to global reduced yield. The core-saturation constraint changed the generated graph, with the same verifier applied to both decoders (Appendix B.3).

A separate three-seed study compared joint denoising with parameter-matched independent precursor roles (FACT). Its full Transformer was trained independently of the production model. Mean exact-L1 products per 1,000 attempts were 856.9 versus 56.9 for Ugi and 767.7 versus 396.4 for aza-Michael addition. Reductive amination favored FACT: 318.8 versus 520.3. Removing cross-attention reduced Ugi and reductive-amination yield but increased aza-Michael yield; removing gradient-conflict control slightly improved all three means. These ablations support reaction-dependent effects, not a uniform benefit from every module.

4.5 GENERATED PRECURSORS EXTENDED BEYOND THE TRAINING CATALOGUES

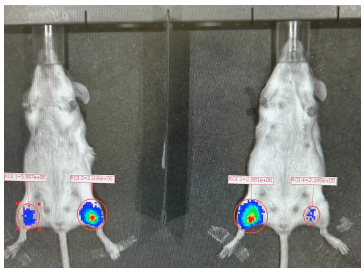
We next asked whether FORGE could preserve exact assembly while generating precursor constitutions absent from training. Under the common Ugi benchmark, the finite-inventory methods produced 1000.0 ± 0.0 verified exact-L1 products per 1,000 attempts but zero component-novel products by construction. FORGE produced 963.9 ± 21.4 verified products and 839.2 ± 26.5 distinct component-novel products per 1,000 attempts (Table 2).

Table 2: **Exact Ugi assembly and component novelty.** Values are mean ± sample standard deviation per 1,000 attempts across three seeds under a common component-disjoint protocol and verifier. Component-novel counts distinct products with one exact decomposition containing a precursor outside the training catalogue; it does not establish strong catalogue escape. †, structural zero; N/E, not estimable.

Models	Valid / 1,000 ↑	Verified L1 / 1,000 ↑	Distinct L1 / 1,000 ↑	Comp.-novel / 1,000 ↑	Held-comp. / 1,000 ↑	Diversity
RGFN	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	$0.0 \pm 0.0^{\dagger}$	0.0 ± 0.0	N/E
DeFoG unconditional	346.5 ± 42.3	280.6 ± 26.6	280.6 ± 26.6	272.0 ± 24.4	15.5 ± 2.4	0.727 ± 0.014
GenMol/SAFE	627.5 ± 6.2	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.873 ± 0.009
Learned inventory selector	1000.0 ± 0.0	1000.0 ± 0.0	849.5 ± 13.7	$0.0 \pm 0.0^{\dagger}$	0.0 ± 0.0	0.717 ± 0.001
Finite catalogue oracle	1000.0 ± 0.0	1000.0 ± 0.0	837.6 ± 8.2	$0.0 \pm 0.0^{\dagger}$	0.0 ± 0.0	0.719 ± 0.000
Shared-null + post-hoc	194.7 ± 15.6	114.4 ± 10.9	108.5 ± 9.7	108.4 ± 9.5	0.4 ± 0.2	0.754 ± 0.014
FACT-matched	60.8 ± 20.2	52.0 ± 15.6	50.8 ± 14.4	49.7 ± 13.0	0.0 ± 0.0	0.667 ± 0.009
FACT-generous	500.7 ± 213.8	296.7 ± 186.0	288.3 ± 175.3	287.8 ± 174.4	0.1 ± 0.2	0.774 ± 0.007
FORGE	965.0 ± 21.9	963.9 ± 21.4	862.6 ± 19.2	839.2 ± 26.5	1.4 ± 1.7	0.643 ± 0.022

Against the fixed training catalogues, component-novel verified exact-L1 yields reached 839.2, 159.3 and 58.7 products per 1,000 attempts across Ugi, aza-Michael addition and reductive amination. The corresponding finite catalogues had zero component-novel yield by construction (Appendix B.4). Representative Ugi products and exact precursors appear in Appendix D.2.

a Mouse bioluminescence image



b Reported 4-hour ROI signals

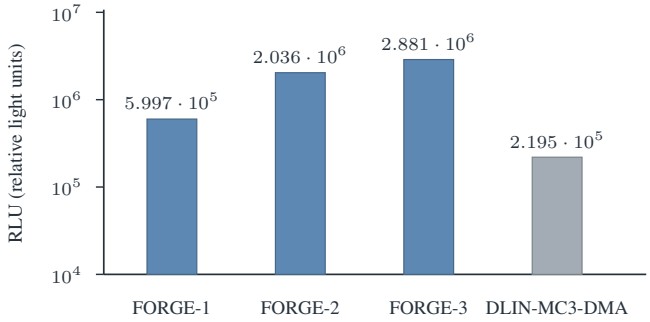


Figure 2: ***In vivo* imaging measurements.** **a**, *In vivo* bioluminescence imaging scan **b**, Single reported 4-hour ROI values on a logarithmic relative light units (RLU) axis from 10^4 to 10^7.

Among the external baselines, DeFoG produced 280.6 ± 26.6 verified products and recovered held components more often than FORGE (15.5 ± 2.4 versus 1.4 ± 1.7 per 1,000 attempts). GenMol/SAFE produced no exact Ugi-L1 product, and RGFN emitted no admissible output. FORGE therefore combined reaction consistency with verifier-relative component novelty, although DeFoG recovered the reserved components more often.

We also compared the generated molecules with known lipids excluded from training. For every method, the two sets differed in their overall structural features. Precision and coverage in fingerprint and descriptor spaces, together with a source-study-grouped classifier two-sample test, found that no method reproduced the full observed-lipid distribution (Appendix B.4). The measured expansion concerns reaction-consistent identities; it does not establish biological utility.

4.6 FORGE LIPIDS DEMONSTRATE STRONG INTRAMUSCULAR POTENCY

To examine reporter expression from experimentally prepared designs, we synthesized three lipids from the shared three-family model, designated FORGE-1, FORGE-2 and FORGE-3, and formulated them into LNPs encapsulating firefly luciferase (FLuc) mRNA. Following intramuscular administration, we observed signals in relative light units (RLU) of 5.997×10^5, 2.036×10^6 and 2.881×10^6, respectively, compared to 2.195×10^5 for the DLIN-MC3-DMA formulation at 4 hrs after treatment (Figure 2). Notably, all three formulations exhibited higher potency than MC3, with FORGE-3 showing a 13.1-fold higher signal. These results help establish that reaction-guided lipids can exhibit high potency *in vivo* and outperform clinical benchmarks. The synthesis and formulation information is given in Appendix B.5.

5 DISCUSSION

In this work, we present FORGE, a generative model that uses known assembly chemistry during whole-molecule generation while leaving precursor identities open. FORGE encodes the intended assembly in a structured reaction program and verifies each accepted product by recovering its precursors and replaying the reaction.

Across three reaction programs (Ugi, repeated aza-Michael addition and repeated reductive amination), FORGE improved exact assembly yield while generating precursor structures absent from the training catalogues. Together, these findings show that assembly chemistry can guide molecular generation without fixing the identities of the reactants in advance. This provides a useful starting point for lipid discovery, where: the model can explore new component structures, and exact replay identifies how the recovered components assemble into each verified product. Those explicit precursor structures also provide targets for availability checks and upstream synthesis planning, connecting molecular design to the practical questions involved in preparing a proposed lipid.

Our experimental results also demonstrate that reaction-guided generation has the potential to achieve strong *in vivo* performance, with our best FORGE lipid outperforming FDA-approved clinical

benchmark lipid MC3. More lipid synthesis, LNP formulation, and biological screening is necessary to further assess whether computational design can translate to reliable delivery activity among a broader set of lipid families. Nonetheless, our current results establish FORGE as a foundation for expanding ionizable-lipid discovery beyond predefined reactant libraries, connecting *de novo* molecular design with verifiable assembly and functional RNA delivery.

AI USE STATEMENT

Generative AI tools assisted with conceptual and methodological development, software implementation and debugging, experimental design and diagnostic analysis, interpretation of computational results, literature search and summarization, and drafting and editing the manuscript. The authors are responsible for the final code, reported results, citations and conclusions, including material produced with AI assistance.

ETHICS STATEMENT

This work evaluates ionizable-lipid generation computationally and includes exploratory mouse bioluminescence measurements with three synthesized FORGE lipids. Molecular design systems can be misused. Chemical validity, exact assembly verification and predicted properties do not establish safety or biological efficacy. The reported mouse measurements do not establish tolerability, tissue selectivity or therapeutic efficacy. Generated compounds without experimental measurements remain computational candidates requiring further evaluation before use.

REPRODUCIBILITY STATEMENT

For reproducibility, we host our code, data-processing scripts, and evaluation pipelines at this anonymous repository:
`https://anonymous.4open.science/r/forge-iclr-review-FBF8/`.

The Methods and Appendix specify the molecular representation, conditional discrete flow, numerical sampler, exact assembly verification, benchmark protocols and evaluation metrics. Appendix A.7 summarizes randomization, historical implementation provenance and the access limitations of the standalone manuscript archive. Seed-level results appear in Appendix B, and assumptions and proofs appear in Appendix C.

REFERENCES

David Adams, Alejandra Gonzalez-Duarte, William D. O'Riordan, Chih-Chao Yang, Mitsuharu Ueda, Arnt V. Kristen, Ivailo Tournev, Hartmut H. Schmidt, Teresa Coelho, John L. Berk, Kon-Ping Lin, Giuseppe Vita, Shahram Attarian, Violaine Planté-Bordeneuve, Michelle M. Mezei, Josep M. Campistol, Juan Buades, Thomas H. Brannagan, Byoung J. Kim, Jeeyoung Oh, Yesim Parman, Yoshiki Sekijima, Philip N. Hawkins, Scott D. Solomon, Michael Polydefkis, Peter J. Dyck, Pritesh J. Gandhi, Sunita Goyal, Jihong Chen, Andrew L. Strahs, Saraswathy V. Nochur, Marianne T. Sweetser, Pushkal P. Garg, Akshay K. Vaishnaw, Jared A. Gollob, and Ole B. Suhr. Patisiran, an RNAi therapeutic, for hereditary transthyretin amyloidosis. *New England Journal of Medicine*, 379(1):11–21, 2018. doi: 10.1056/NEJMoa1716153.

Akin Akinc, Andreas Zumbuehl, Michael Goldberg, Elizaveta S. Leshchiner, Valentina Busini, Naushad Hossain, Sergio A. Bacallado, David N. Nguyen, Jason Fuller, Rene Alvarez, Anna Borodovsky, Todd Borland, Rainer Constien, Antonin de Fougerolles, J. Robert Dorkin, K. Narayanannair Jayaprakash, Muthusamy Jayaraman, Matthias John, Victor Koteliansky, Muthiah Manoharan, Lubomir Nechev, June Qin, Timothy Racie, Denitza Raitcheva, Kallanthottathil G. Rajeev, Dinah W. Y. Sah, Jürgen Soutschek, Ivanka Toudjarska, Hans-Peter Vornlocher, Tracy S. Zimmermann, Robert Langer, and Daniel G. Anderson. A combinatorial library of lipid-like materials for delivery of RNAi therapeutics. *Nature Biotechnology*, 26(5):561–569, 2008. doi: 10.1038/nbt1402.

Lindsey R. Baden, Hana M. El Sahly, Brandon Essink, Karen Kotloff, Sharon Frey, Rick Novak, David Diemert, Stephen A. Spector, Nadine Rouphael, C. Buddy Creech, John McGettigan, Shishir Khetan, Nathan Segall, Joel Solis, Adam Brosz, Carlos Fierro, Howard Schwartz, Kathleen Neuzil, Lawrence Corey, Peter Gilbert, Holly Janes, Dean Follmann, Mary Marovich, John Mascola, Laura Polakowski, Julie Ledgerwood, Barney S. Graham, Hamilton Bennett, Rolando Pajon, Conor Knightly, Brett Leav, Weiping Deng, Honghong Zhou, Shu Han, Melanie Ivarsson, Jacqueline Miller, and Tal Zaks. Efficacy and safety of the mRNA-1273 SARS-CoV-2 vaccine. *New England Journal of Medicine*, 384(5):403–416, 2021. doi: 10.1056/NEJMoa2035389.

Jingan Chen, Yue Xu, Muye Zhou, Shufen Xu, Andrew James Varley, Alex Golubovic, Rick Xing Ze Lu, Kevin Chang Wang, Mina Yeganeh, Daniel Vosoughi, and Bowen Li. Combinatorial design of ionizable lipid nanoparticles for muscle-selective mRNA delivery with minimized off-target effects. *Proceedings of the National Academy of Sciences*, 120(50):e2309472120, 2023. doi: 10.1073/pnas.2309472120. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC10723144/.

Qiang Cheng, Tuo Wei, Lukas Farbiak, Lindsay T. Johnson, Sean A. Dilliard, and Daniel J. Siegwart. Selective organ targeting (SORT) nanoparticles for tissue-specific mRNA delivery and CRISPR-Cas gene editing. *Nature Nanotechnology*, 15(4):313–320, 2020. doi: 10.1038/s41565-020-0669-6.

Miruna Cretu, Charles Harris, Ilia Igashov, Arne Schneuing, Marwin Segler, Bruno Correia, Julien Roy, Emmanuel Bengio, and Pietro Liò. SynFlowNet: Design of diverse and novel molecules with synthesis constraints. In *International Conference on Learning Representations*, 2025.

Pieter R. Cullis and Michael J. Hope. Lipid nanoparticle systems for enabling gene therapies. *Molecular Therapy*, 25(7):1467–1475, 2017. doi: 10.1016/j.ymthe.2017.03.013.

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

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

Julian D. Gillmore, Ed Gane, Jorg Taubel, Justin Kao, Marianna Fontana, Michael L. Maitland, Jessica Seitzer, Daniel O'Connell, Kathryn R. Walsh, Kristy Wood, Jonathan Phillips, Yuanxin Xu, Adam Amaral, Adam P. Boyd, Jeffrey E. Cehelsky, Mark D. McKee, Andrew Schiermeier, Olivier Harari, Andrew Murphy, Christos A. Kyratsous, Brian Zambrowicz, Randy Soltys, David E. Gutstein, John Leonard, Laura Sepp-Lorenzino, and David Lebwohl. CRISPR-Cas9 *in vivo* gene editing for transthyretin amyloidosis. *New England Journal of Medicine*, 385(6):493–502, 2021. doi: 10.1056/NEJMoa2107454.

Xuexiang Han, Hanwen Zhang, Kamila Butowska, Kelsey L. Swingle, Mohamad-Gabriel Alameh, Drew Weissman, and Michael J. Mitchell. An ionizable lipid toolbox for RNA delivery. *Nature Communications*, 12:7233, 2021. doi: 10.1038/s41467-021-27493-0.

Marine Z. C. Hatit, Curtis N. Dobrowolski, Melissa P. Lokugamage, David Loughrey, Huanzhen Ni, Chiara Zurla, Alejandro J. Da Silva Sanchez, Afsane Radmand, Sebastian G. Huayamares, Ryan Zenhausern, Kalina Paunovska, Hannah E. Peck, Jinwhan Kim, Manaka Sato, Jacob I. Feldman, Michael-Alexander Rivera, Ana Cristian, YongTae Kim, Philip J. Santangelo, and James E. Dahlman. Nanoparticle stereochemistry-dependent endocytic processing improves *in vivo* mRNA delivery. *Nature Chemistry*, 15(4):508–515, 2023. doi: 10.1038/s41557-023-01138-9.

Xucheng Hou, Tal Zaks, Robert Langer, and Yizhou Dong. Lipid nanoparticles for mRNA delivery. *Nature Reviews Materials*, 6(12):1078–1094, 2021. doi: 10.1038/s41578-021-00358-0.

Muthusamy Jayaraman, Steven M. Ansell, Barbara L. Mui, Ying K. Tam, Jianxin Chen, Xinyao Du, David Butler, Laxman Eltepu, Shigeo Matsuda, Jayaprakash K. Narayanannair, Kallanthottathil G. Rajeev, Ismail M. Hafez, Akin Akinc, Martin A. Maier, Mark A. Tracy, Pieter R. Cullis, Thomas D. Madden, Muthiah Manoharan, and Michael J. Hope. Maximizing the potency of siRNA lipid nanoparticles for hepatic gene silencing *in vivo*. *Angewandte Chemie International Edition*, 51(34):8529–8533, 2012. doi: 10.1002/anie.201203263.

Michał Koziarski, Andrei Rekesh, Dmytro Shevchuk, Almer van der Sloot, Piotr Gaiński, Yoshua Bengio, Cheng-Hao Liu, Mike Tyers, and Robert Batey. RGFN: Synthesizable molecular generation using GFlowNets. In *Advances in Neural Information Processing Systems*, volume 37, pp. 46908–46955, 2024.

Seul Lee, Karsten Kreis, Srimukh Prasad Veccham, Meng Liu, Danny Reidenbach, Yuxing Peng, Saee Gopal Paliwal, Weili Nie, and Arash Vahdat. GenMol: A drug discovery generalist with discrete diffusion. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 33205–33226. PMLR, 2025.

Bowen Li, Rajith Singh Manan, Shun-Qing Liang, Akiva Gordon, Allen Jiang, Andrew Varley, Guangping Gao, Robert Langer, Wen Xue, and Daniel Anderson. Combinatorial design of nanoparticles for pulmonary mRNA delivery and genome editing. *Nature Biotechnology*, 41(10): 1410–1415, 2023. doi: 10.1038/s41587-023-01679-x.

Bowen Li, Idris O. Raji, Akiva G. R. Gordon, Lizhuang Sun, Theresa M. Raimondo, Favour A. Oladimeji, Allen Y. Jiang, Andrew Varley, Robert S. Langer, and Daniel G. Anderson. Accelerating ionizable lipid discovery for mRNA delivery using machine learning and combinatorial chemistry. *Nature Materials*, 23(7):1002–1008, 2024. doi: 10.1038/s41563-024-01867-3.

Kevin T. Love, Kerry P. Mahon, Christopher G. Levins, Kathryn A. Whitehead, William Querbes, J. Robert Dorkin, June Qin, William Cantley, Liu Liang Qin, Timothy Racie, Maria Frank-Kamenetsky, Ka Ning Yip, Rene Alvarez, Dinah W. Y. Sah, Antonin de Fougerolles, Kevin Fitzgerald, Victor Koteliansky, Akin Akinc, Robert Langer, and Daniel G. Anderson. Lipid-like materials for low-dose, *in vivo* gene silencing. *Proceedings of the National Academy of Sciences*, 107(5):1864–1869, 2010. doi: 10.1073/pnas.0910603106.

Rohin Maganti, Rahul Maganti, and Mohamad-Gabriel Alameh. Synthesis-constrained discrete diffusion for ionizable lipid generation. In *ICLR 2026 Workshop on Generative and Experimental Perspectives for Biomolecular Design (GEM)*, 2026. URL https://openreview.net/forum?id=6RFQqfjD06.

Lei Miao, Linxian Li, Yuxuan Huang, Derfogail Delcassian, Jasdave Chahal, Jinsong Han, Yunhua Shi, Kaitlyn Sadtler, Wenting Gao, Jiaqi Lin, Joshua C. Doloff, Robert Langer, and Daniel G. Anderson. Delivery of mRNA vaccines with heterocyclic lipids increases anti-tumor efficacy by STING-mediated immune cell activation. *Nature Biotechnology*, 37(10):1174–1185, 2019. doi: 10.1038/s41587-019-0247-3.

Yuxuan Ou, Jingyi Zhao, Austin Tripp, Morteza Rasoulianboroujeni, and José Miguel Hernández-Lobato. A deep generative model for the design of synthesizable ionizable lipids. In *NeurIPS 2024 Workshop on AI for New Drug Modalities*, 2024. arXiv:2412.00928.

Fernando P. Polack, Stephen J. Thomas, Nicholas Kitchin, Judith Absalon, Alejandra Gurtman, Stephen Lockhart, John L. Perez, Gonzalo Pérez Marc, Edson D. Moreira, Cristiano Zerbini, Ruth Bailey, Kena A. Swanson, Satrajit Roychoudhury, Kenneth Koury, Ping Li, Warren V. Kalina, David Cooper, Robert W. Frenck, Laura L. Hammitt, Özlem Türeci, Haylene Nell, Axel Schaefer, Serhat Ünal, Dina B. Tresnan, Susan Mather, Philip R. Dormitzer, Uğur Şahin, Kathrin U. Jansen, and William C. Gruber. Safety and efficacy of the BNT162b2 mRNA Covid-19 vaccine. *New England Journal of Medicine*, 383(27):2603–2615, 2020. doi: 10.1056/NEJMoa2034577.

Yiming Qin, Manuel Madeira, Dorina Thanou, and Pascal Frossard. DeFoG: Discrete flow matching for graph generation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50269–50326. PMLR, 2025.

Andrei Rekesh, Miruna Cretu, Dmytro Shevchuk, Vignesh Ram Somnath, Pietro Liò, Robert A. Batey, Mike Tyers, Michał Koziarski, and Cheng-Hao Liu. SynCoGen: Synthesizable 3D molecule generation via joint reaction and coordinate modeling. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=24QKU4iqft.

Sean C. Semple, Akin Akinc, Jianxin Chen, Ammen P. Sandhu, Barbara L. Mui, Connie K. Cho, Dinah W. Y. Sah, Derrick Stebbing, Erin J. Crosley, Ed Yaworski, Ismail M. Hafez, J. Robert Dorkin, June Qin, Kieu Lam, Kallanthottathil G. Rajeev, Kim F. Wong, Lloyd B. Jeffs, Lubomir Nechev, Merete L. Eisenhardt, Muthusamy Jayaraman, Mikameh Kazem, Martin A. Maier, Masuna Srinivasulu, Michael J. Weinstein, Qingmin Chen, Rene Alvarez, Scott A. Barros, Soma De, Sandra K. Klimuk, Todd Borland, Verbena Kosovrasti, William L. Cantley, Ying K. Tam, Muthiah Manoharan, Marco A. Ciufolini, Mark A. Tracy, Antonin de Fougerolles, Ian MacLachlan, Pieter R. Cullis, Thomas D. Madden, and Michael J. Hope. Rational design of cationic lipids for siRNA delivery. *Nature Biotechnology*, 28(2):172–176, 2010. doi: 10.1038/nbt.1602.

Yue Xu, Shihao Ma, Haotian Cui, Jingan Chen, Shufen Xu, Fanglin Gong, Alex Golubovic, Muye Zhou, Kevin Chang Wang, Andrew Varley, Rick Xing Ze Lu, Bo Wang, and Bowen Li. AGILE platform: A deep learning powered approach to accelerate LNP development for mRNA delivery. *Nature Communications*, 15:6305, 2024. doi: 10.1038/s41467-024-50619-z.

Lulu Xue, Alex G. Hamilton, Gan Zhao, Zebin Xiao, Rakan El-Mayta, Xuexiang Han, Ningqiang Gong, Xinhong Xiong, Junchao Xu, Christian G. Figueroa-Espada, Sarah J. Shepherd, Alvin J. Mukalel, Mohamad-Gabriel Alameh, Jiaxi Cui, Karin Wang, Andrew E. Vaughan, Drew Weissman, and Michael J. Mitchell. High-throughput barcoding of nanoparticles identifies cationic, degradable lipid-like materials for mRNA delivery to the lungs in female preclinical models. *Nature Communications*, 15(1):1884, 2024. doi: 10.1038/s41467-024-45422-9.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

A METHOD AND IMPLEMENTATION DETAILS

This appendix specifies program and graph encoding, model training, sampling and exact assembly verification. Algorithm 1 combines the inference steps.

A.1 REACTION-PROGRAM ENCODING

The program tuple in Equation 1 uses the ordered atom-position set $\mathcal{I}_{\mathrm{atom}}(P) = \{1, \ldots, n(P)\}$. The arrays $\mathbf{r}$ and $\mathbf{k}$ assign precursor roles and core positions to these atoms. This atom-position index differs from $\mathcal{I}(P)$, which indexes all six families of categorical coordinates.

Morphology is computed from the selected atom-mapped trace and its sparse serialization. For role r, let V_r^{ext} contain its exterior atoms, excluding reaction-core atoms, and let $o_r(v)$ count the same-role exterior children of v in the serialized tree. The four fields are

$$M_r = (n_r, j_r, \gamma_r, a_r), \qquad n_r = |V_r^{\mathrm{ext}}|, \qquad j_r = \sum_{v \in V_r^{\mathrm{ext}}} \max\{o_r(v) - 1, 0\}. \tag{10}$$

Here γ_r counts closure edges with both endpoints in V_r^{ext}, and a_r counts same-role bonds between exterior and core atoms. When the serialized tree restricts to a spanning forest of the exterior subgraph, γ_r equals that subgraph's cycle rank. The junction budget is a tree-branching count; it does not count degree contributed by closure bonds. Consequently these fields depend on the selected annotation and serialization, not on the unannotated graph alone. Fixed-coordinate equality alone does not establish agreement with every morphology field. Positions without precursor origin use the designated absent morphology state. Integer morphology values are encoded with an offset of one, reserving zero for that absent state.

The training-fold program prior is defined in Section 3.2. At graph position i, the program embedding is

$$h_i^P = \mathrm{LN}\left(e_c(c) + e_d(d) + e_r(r_i) + e_k(k_i) + \sum_{u=1}^{4} e_{M,u}(M_{r_i,u}) \right). \tag{11}$$

Here, e_c, e_d, e_r, e_k and $e_{M,u}$ are learned embeddings in a shared hidden space, and LN denotes layer normalization. The embedding encodes precursor role, reaction-core status and role-local morphology. The morphology fields determine active positions and fixed or variable masks. Sampling from the prior specifies these constraints; generation assigns the atom, bond and topology states within each role.

A chemistry-specific adapter determines the precursor roles, reaction core, reverse decomposition and forward transform associated with c. Its sampling layout determines active positions and fixed or variable coordinates; the embeddings condition neural prediction on the remaining program fields.

A.2 GRAPH SERIALIZATION AND SOURCE ANNOTATIONS

A spanning-tree representation with at most three residual closure edges preserves the full 194-atom support without an $O(N_{\max}^2)$ edge state. The discrete flow generates the atom, bond and topology states in this representation. Let $\alpha = (\mathbf{C}, \tau, \mathbf{r}, \mathbf{k})$ contain the selected exact precursor tuple, registered assembly trace, atom-wise precursor-role assignment and reaction-core annotation. The deterministic program-aware encoder

$$\mathrm{Enc}_P : \{(G, \alpha) : G \in \mathfrak{G}_{N_{\max}, K_{\mathrm{cl}}} \text{ and } \alpha \text{ is admitted for } P\} \longrightarrow \mathfrak{X}_{N_{\max}, K_{\mathrm{cl}}} \tag{12}$$

constructs the categorical graph state

$$X_\star := \mathrm{Enc}_P(G, \alpha) = \left(\mathbf{a}, \boldsymbol{\pi}, \mathbf{b}, \mathbf{e}^L, \mathbf{e}^R, \mathbf{z}\right) \tag{13}$$

where $\mathbf{a}$ contains atom states, $\boldsymbol{\pi}$ parent pointers, $\mathbf{b}$ spanning-tree bond states, $\mathbf{e}^L$ and $\mathbf{e}^R$ closure-endpoint pointers, and $\mathbf{z}$ closure-bond states.

The encoder first partitions the atoms into connected precursor-origin components under α. It selects as root the reaction-core atom with the smallest tie-broken canonical RDKit rank, and orders the origin-component graph by breadth-first traversal from that root component, breaking ties by role index,

component canonical ranks and original atom index. Within each component it uses a deterministic depth-first preorder from the selected attachment atom, with neighbors ordered by the same canonical ranks. The traversed bonds define the spanning tree. Every residual bond is represented once as an ordered endpoint pair with its bond label, and residual triples are sorted lexicographically. Role, core, fixed-coordinate and morphology arrays follow the same atom order. We excluded training records with more than one qualified exact source trace from semantic training and collapsed duplicate descriptions of an admitted trace to its canonical record. Thus $\mathrm{Dec}(\mathrm{Enc}_P(G, \alpha)) \cong G$ for every admitted record.

The raw padded categorical product space $\overline{\mathfrak{X}}_{N_{\max},K_{\mathrm{cl}}}$ also contains malformed states. Its well-formed subset $\mathfrak{X}_{N_{\max},K_{\mathrm{cl}}}$ enforces an active node prefix, earlier-parent pointers, distinct valid closure endpoints, no duplicate edges, and null values in unused closure slots. The total decoder, including its failure value, is typed as

$$\mathrm{Dec} : \overline{\mathfrak{X}}_{N_{\max},K_{\mathrm{cl}}} \longrightarrow \mathfrak{G}_{N_{\max},K_{\mathrm{cl}}} \cup \{\perp_{\mathrm{dec}}\}. \tag{14}$$

For this study, $N_{\max} = 194$ and $K_{\mathrm{cl}} = 3$. These bounds specify architectural support. Reaction consistency, nonzero probability under a trained model and exact assembly require separate conditions.

The structural decoder specifies graph reconstruction from an already chosen state. The production terminal readout additionally chooses that state from neural predictions (Appendix A.4). These are separate operations.

The chemistry adapter marks any atom, pointer, or bond states fixed by P. FORGE holds these states constant while corrupting the remaining states along the linear probability path and predicting their clean values from (X_t, t, P). Let Ω_P be the subset of the raw state space satisfying the allowed coordinate alphabets and fixed values, and define the admissible program set

$$\mathcal{P}_{\mathrm{adm}} = \{P : \mathfrak{X}_{N_{\max},K_{\mathrm{cl}}} \cap \Omega_P \neq \varnothing\}. \tag{15}$$

The morphology prior is normalized only on $\mathcal{P}_{\mathrm{adm}}$. Each chemistry adapter is required to satisfy the program-compatible encoding property: every admitted annotated record obeys $\mathrm{Enc}_P(G, \alpha) \in \Omega_P$. During corpus construction, we checked this property and exact constitutional decode round trip for every record before split assignment. Records failing either check were excluded. The checks covered all cached training, calibration and held-out records; they do not establish the property for unregistered chemistry.

A.3 Transformer architecture and training objectives

The denoiser is a graph Transformer over the complete sparse state. Each layer applies graph self-attention with relation biases for parent, child and closure edges, then cross-attends to memory tokens representing program identity, program depth, precursor role and reaction-core position. Cross-attention is repeated in every layer so that reaction semantics can influence both local state updates and long-range coordination among precursor-derived regions. Three softly program-routed residual adapters add family-specific parameters to the shared whole-graph backbone.

The reported final three-family model used a stratified minibatch with 21 records per chemistry and source-weighted sampling within each stratum. Two such microbatches formed an optimizer update. Training times were $t = \min\{0.98, \max\{0.02, U\}\}$ for $U \sim \mathcal{U}(0, 1)$, giving point masses of 0.02 at both endpoints and unit density on $(0.02, 0.98)$. Thus training did not cover all times required by the ideal reference objective in Equation 7.

For chemistry c and state family $q \in \mathscr{J} = \{\mathbf{a}, \boldsymbol{\pi}, \mathbf{b}, \mathbf{e}^L, \mathbf{e}^R, \mathbf{z}\}$, let $\mathcal{S}_{c,q}$ be the selected variable coordinates across that chemistry's minibatch. With coordinate loss $\ell_i = -\log d_{\theta,i}(X_{\star,i} \mid X_t, t, P)$, the pooled reduction is

$$L^{\mathrm{pool}}_{c,q} = \frac{\sum_{i \in \mathcal{S}_{c,q}} \ell_i}{\max\{1, |\mathcal{S}_{c,q}|\}}. \tag{16}$$

Coordinates here include their example index. Chemistry states (atoms, parent bonds and closure bonds) additionally use equal mass over roles present in the selected coordinates. If $\mathcal{S}_{c,q,r}$ contains coordinates assigned to role r and $\mathcal{R}_{c,q} = \{r : |\mathcal{S}_{c,q,r}| > 0\}$, this reduction is

$$L^{\mathrm{role}}_{c,q} = \frac{1}{|\mathcal{R}_{c,q}|} \sum_{r \in \mathcal{R}_{c,q}} \frac{\sum_{i \in \mathcal{S}_{c,q,r}} \ell_i}{|\mathcal{S}_{c,q,r}|}, \tag{17}$$

with zero loss if no role is present. Atom and parent-bond groups use their position's role; closure-bond groups use the role at the clean left endpoint. Pointer losses use the pooled reduction and mask logits to the permitted layout positions. The flow term $L_{c,\mathrm{flow}}$ sums these six reduced losses, with unit weights.

For each chemistry, the recorded task loss was

$$\begin{aligned} L_c = L_{c,\mathrm{flow}} + 0.25L_{c,\mathrm{role}} + 0.25L_{c,\mathrm{core}} + 0.25L_{c,\mathrm{repeat}} \\ + L_{c,\mathrm{child}} + 0.5L_{c,\mathrm{junction}} + L_{c,\mathrm{chem|top}}. \end{aligned} \tag{18}$$

The role and core terms are cross-entropies averaged equally over present target states. The repeat term averages squared differences between predicted atom distributions, and separately parent-bond distributions, at aligned exterior positions in repeated component instances; it sums the two means. The child term is state-balanced cross-entropy for same-component exterior child counts in $\{0, 1, 2, 3\}$. The junction term applies smooth-L1 loss to predicted and target role-local junction totals, each divided by the number of exterior atoms in that role, then averages over eligible examples within a role and equally across present roles. Predicted junction totals sum the expected child-count contribution $\max\{o - 1, 0\}$ across exterior atoms. The topology-conditioned chemistry term repeats the role-balanced atom and bond losses in a second forward pass with clean parent and closure endpoints and corrupted atom and bond states. These definitions describe the reported final arm; the earlier architecture study used its own recorded ablation settings.

PCGrad combines the task gradients in integer chemistry-state order. For $g_c = \nabla_\theta L_c$, it initializes $\widetilde{g}_c = g_c$ and, for each other task h in that order, applies

$$\widetilde{g}_c \leftarrow \widetilde{g}_c - \mathbf{1}\{\langle \widetilde{g}_c, g_h \rangle < 0\} \frac{\langle \widetilde{g}_c, g_h \rangle}{\max\{\|g_h\|_2^2, 10^{-12}\}} g_h \tag{19}$$

on parameter coordinates available to both tasks. Gradients are not norm-normalized, and the optimizer receives the arithmetic mean of the projected task gradients. This update is not generally the gradient of a scalar sum of task losses. No population optimality claim is made for the full procedure. In particular, target-dependent group reductions and the restricted training-time range require analysis beyond Proposition C.7.

We tested layerwise cross-attention, each auxiliary loss, routed adapters and gradient-conflict control through separate interventions (Appendix B.3).

A.4 Numerical sampling and terminal decoding

The finite sampler evaluates the denoiser at $t_k = k/L_{\mathrm{samp}}$, for $k = 0, \ldots, L_{\mathrm{samp}} - 1$, and uses $\Delta t = 1/L_{\mathrm{samp}}$. Conditional on $Z_k = x$, it draws one clean category $A_{k,i} \sim d_{\theta,i}(\cdot \mid x, t_k, P)$ independently for each variable coordinate. Write $s^P_{t,i}(u \mid a) = (1 - t)q^P_{0,i}(u) + t\mathbf{1}\{u = a\}$ for its conditional path factor, $\dot{s}^P_i(u \mid a) = \mathbf{1}\{u = a\} - q^P_{0,i}(u)$ for the derivative, and $[z]_+ = \max\{z, 0\}$. The code uses the conditional off-diagonal rate

$$R^{*,\varepsilon}_{t,i}(u, v \mid a, P) = \frac{[\dot{s}^P_i(v \mid a) - \dot{s}^P_i(u \mid a)]_+}{|\mathcal{X}_i(P)| \max\{s^P_{t,i}(u \mid a), \varepsilon\}}, \quad u \neq v, \qquad \varepsilon = 10^{-8}. \tag{20}$$

Full source support is required only on the allowed coordinate alphabet, including the masked candidate set for a pointer. Thus that alphabet is the support for every $t_k < 1$. The denominator clamp in Equation 20 is a numerical safeguard absent from the ideal rate in Appendix C.3.

For the sampled endpoint $a = A_{k,i}$, let $h_{k,i} = \Delta t \sum_{v \neq x_i} R^{*,\varepsilon}_{t_k,i}(x_i, v \mid a, P)$ and set $\kappa_{k,i} = \min\{1, 0.999/h_{k,i}\}$ when $h_{k,i} > 0$, with $\kappa_{k,i} = 1$ otherwise. The implemented coordinate kernel is

$$\widehat{K}_{k,i}(v \mid x, a) = \begin{cases} \kappa_{k,i}\Delta t\, R^{*,\varepsilon}_{t_k,i}(x_i, v \mid a, P), & v \neq x_i, \\ 1 - \kappa_{k,i}h_{k,i}, & v = x_i. \end{cases} \tag{21}$$

Its off-diagonal masses sum to at most 0.999; the diagonal is the remaining mass. Consequently all entries are nonnegative and each row sums to one. Coordinates are drawn independently from these kernels, followed by the projection Π_P.

Without the clamp or cap, and when every conditional Euler row is nonnegative, averaging over $A_{k,i}$ gives the Euler kernel built from Equation 6 exactly. Sampling one endpoint per coordinate preserves

the transition probabilities of the uncapped update. The nonlinear cap generally changes the averaged kernel. Independent coordinate updates also permit simultaneous finite-step changes, so their product is a numerical approximation to the local CTMC, not its exact transition kernel. No discretization error bound is asserted near the terminal singularity.

After all steps, the model evaluates $(Z_{L_{\mathrm{samp}}}, 1, P)$ once more. The production terminal readout H_P^θ selects coordinate values using constrained argmax, restores fixed states and returns either a raw state or a typed terminal failure, denoted collectively by $\perp_{\mathrm{term}}$. For Ugi it enforces one hydrogen on the aldehyde-derived α carbon and one on the isocyanide-derived amide nitrogen. The selected state is then reconstructed and checked for connectedness, valence and sanitization. A readout failure retains its detailed reason in the ledger; for graph-law notation we extend $\mathrm{Dec}(\perp_{\mathrm{term}}) = \perp_{\mathrm{dec}}$. If $q^P_{\theta,\mathrm{out}}$ is the law after terminal readout, including failure, then

$$p_\theta(\cdot \mid P) = \mathrm{Dec}_{\#}\, q^P_{\theta,\mathrm{out}}, \qquad p_\theta(G \mid P) = \sum_{z:\mathrm{Dec}(z)=G} q^P_{\theta,\mathrm{out}}(z). \tag{22}$$

This law retains structural failures and chemically invalid graphs before the validity and L1 filters; it is not conditioned on acceptance. The ideal theorem does not include the learned $t = 1$ readout, which is unconstrained by a loss integrated over $t \in (0, 1)$. We compared core-saturation and valence-only readouts under the same trained weights and verifier, then separately compared program-role and global source marginals (Appendix B.3).

A.5 Exact assembly verification

For the chemistry c specified by P, FORGE decomposes each valid graph into role-assigned precursors and applies the registered forward transform. Exact replay checks assembly existence under c; it does not test full agreement with the sampled conditioning layout.

Definition A.1 (Exact assembly and recovered witnesses)**.** For the registered forward operator $\mathcal{F}_c$ on precursor inputs $\mathfrak{C}_c$, define

$$\mathcal{E}^\star_c(G) = \{\mathbf{C} \in \mathfrak{C}_c : G \in \mathcal{F}_c(\mathbf{C})\}, \qquad \mathrm{L1}^\star(G, c) = \mathbf{1}\{\mathcal{E}^\star_c(G) \neq \varnothing\}. \tag{23}$$

The bounded reverse procedure returns candidate pairs $(\mathbf{C}, \tau)$ in $\mathcal{D}_c(G)$. Equation 2 retains the pairs passing exact forward replay, and

$$\widehat{\mathrm{L1}}(G, c) = \mathbf{1}\{\widehat{\mathcal{E}}_c(G) \neq \varnothing\}. \tag{24}$$

Both predicates describe assembly existence; molecular validity is checked separately.

The set $\mathcal{E}^\star_c(G)$ contains precursor inputs, whereas $\widehat{\mathcal{E}}_c(G)$ contains input–trace pairs. Their compatible comparison is

$$\{\mathbf{C} : \exists \tau,\ (\mathbf{C}, \tau) \in \widehat{\mathcal{E}}_c(G)\} \subseteq \mathcal{E}^\star_c(G), \qquad \widehat{\mathrm{L1}}(G, c) \leq \mathrm{L1}^\star(G, c). \tag{25}$$

The inclusion follows directly from exact replay. Its converse requires complete reverse search, which has not been established. Throughout the paper, verified exact-L1 yield counts valid attempts satisfying the recovered predicate. Candidate traces are compared after the registered role- and step-aware canonicalization, so duplicate encodings do not create additional witnesses.

Post-hoc filtering changes which completed samples are retained but cannot change the proposal distribution's exact-L1 acceptance probability. Under independent sampling, this probability determines the complete distribution of accepted counts under a fixed attempt budget and the expected cost of obtaining a fixed number of accepted products (Proposition C.10). We distinguish decomposition coverage from forward-replay precision because a candidate disconnection may fail to reconstruct G. Candidate decomposition coverage is the fraction of valid graphs for which $\mathcal{D}_c(G) \neq \varnothing$. For a declared sequence of valid evaluation graphs $G_1, \ldots, G_m$, candidate replay precision is

$$\mathrm{ReplayPrecision} = \frac{\sum_{b=1}^m |\widehat{\mathcal{E}}_c(G_b)|}{\sum_{b=1}^m |\mathcal{D}_c(G_b)|}, \tag{26}$$

with N/E when the denominator is zero. Replay of tuples already in $\widehat{\mathcal{E}}_c(G)$ has 100% success by construction. This invariant does not estimate candidate replay precision. We separately define

$$\mathrm{CandidateAmbig}(G, c) = \mathbf{1}\{|\mathcal{D}_c(G)| > 1\}, \qquad \mathrm{ExactAmbig}(G, c) = \mathbf{1}\{|\widehat{\mathcal{E}}_c(G)| > 1\}. \tag{27}$$

A distinct exact-L1 product is a deduplicated generated constitution. An unambiguous exact decomposition satisfies $|\widehat{\mathcal{E}}_c(G)| = 1$. We distinguish product uniqueness from decomposition uniqueness throughout.

A.6 End-to-end inference procedure

Algorithm 1 records every generation attempt, including failures in molecular validation and exact replay. All verified precursor traces are retained; multiple exact decompositions are recorded as ambiguity rather than excluded from exact-L1 yield.

Algorithm 1 FORGE generation and exact assembly verification.

Require: trained coordinate predictors, chemistry c, program prior $p_{\mathrm{prog}}(\cdot \mid c)$, integers $L_{\mathrm{samp}} \geq 1$ and $B \geq 0$, recorded seed s
Ensure: attempt-indexed exact-L1 ledger $\mathcal{L}_{\mathrm{out}}$ and typed failure ledger $\mathcal{H}$

initialize the random-number streams from s; $\mathcal{L}_{\mathrm{out}} \leftarrow [\,]$; $\mathcal{H} \leftarrow [\,]$
for $b = 1, \ldots, B$ **do**
  sample and record $P \sim p_{\mathrm{prog}}(\cdot \mid c)$
  sample $Z_0 \sim q_0^P$ and set $Z \leftarrow \Pi_P(Z_0)$
  for $k = 0, \ldots, L_{\mathrm{samp}} - 1$ **do**
    $t_k \leftarrow k/L_{\mathrm{samp}}$; $\Delta t_k \leftarrow 1/L_{\mathrm{samp}}$
    $Z \leftarrow \Pi_P(\textsc{CappedCoordinateEuler}(Z, t_k, \Delta t_k, P))$
  end for
  $Z_{\mathrm{out}} \leftarrow H_P^\theta(Z)$ ▷ fresh prediction at $t = 1$, constrained readout
  if Z_{out} is a terminal failure **then**
    record $(b, \textsc{TerminalFailure}, \text{readout reason})$ in $\mathcal{H}$; **continue**
  end if
  $G \leftarrow \mathrm{Dec}(Z_{\mathrm{out}})$
  if $G = \perp_{\mathrm{dec}}$ or $\mathrm{Valid}(G) = 0$ **then**
    record $(b, \textsc{InvalidGraph}, \text{decode or validation reason})$ in $\mathcal{H}$
    continue
  end if
  $\widehat{\mathcal{E}}_c(G) \leftarrow \{(\mathbf{C}, \tau) \in \mathcal{D}_c(G) : \exists H \in \mathcal{F}_c(\mathbf{C}),\ H \cong G\}$
  if $\widehat{\mathcal{E}}_c(G) = \varnothing$ **then**
    $\mathcal{H} \leftarrow \mathcal{H} \,\|\, [(b, \textsc{L1Failure})]$
    continue
  end if
  $\mathcal{L}_{\mathrm{out}} \leftarrow \mathcal{L}_{\mathrm{out}} \,\|\, [(b, G, \widehat{\mathcal{E}}_c(G))]$
end for
return $\mathcal{L}_{\mathrm{out}}, \mathcal{H}$

A.7 Reproducibility and artifact provenance

The computational results use the historical shared three-family model. Training-seed labels 0, 1 and 2 correspond to random-number seeds 20260825, 20260826 and 20260827, respectively (Table 4). Each comparison retains its declared training split, attempt budget, checkpoint rule and evaluator. The manuscript provenance records identify the source artifacts for the reported tables and figures through SHA-256 hashes.

The historical source verification matched the complete seed-0 evaluation-source fingerprint to a retained code snapshot. No exact match was recovered for the seed-1 and seed-2 evaluation-source fingerprint among the retained snapshots inspected, and a complete training-source manifest was not recovered. Current development code does not establish the historical implementation used for those runs.

The standalone manuscript archive contains the LaTeX source, generated tables, figure assets and provenance records. It does not bundle the training corpora, model checkpoints, full evaluation ledgers or a complete historical training-environment and compute manifest. Independent end-to-end reruns therefore require access to the corresponding historical run artifacts. The supplied experimental images and ROI values are included; the unresolved experimental details are described in Section 4.6 and Appendix B.5.

B EVALUATION PROTOCOL AND ADDITIONAL RESULTS

The evaluation protocol applies to Section 4. Additional analyses quantify decomposition outcomes, variation across training seeds, component diversity and the bounded guidance diagnostics. Each comparison includes its seed-level results and metric definitions.

B.1 EVALUATION PROTOCOL

B.1.1 DATA, SPLITS AND GENERATION BUDGETS

We evaluated three assembly programs with different amounts of supervision: Ugi, repeated aza-Michael addition and repeated reductive amination. The comparisons tested shared generation, reaction conditioning and recovery of precursors absent from training. We trained three matched shared-model arms: correct program conditioning, no program information and cyclically incorrect reaction labels. All arms used the same architecture and optimization budget. We evaluated the prespecified final checkpoint from three independent training seeds under equal generation-attempt budgets per program. Our primary metric is verified exact-L1 yield per generation attempt: the number of generated graphs that pass molecular validity, program-aware decomposition, and exact forward replay divided by the total number of sampling attempts. Secondary metrics separate graph quality (validity and connectedness), reaction recovery (verified coverage, the replay invariant, abstention, and exact ambiguity), and exploration (distinct-product fraction, whole-product and component novelty, molecular diversity, and effective component count). We report exact counts, means across seeds, sample standard deviations, individual seed values, and the minimum and maximum of the three paired seed contrasts. These are descriptive across-seed summaries, not confidence intervals. Generated molecules are not treated as independent training replicates. Conditional binomial variation across a fixed model's 3,072 attempts is distinct from variation induced by retraining. Training used only the component-disjoint training folds. We used calibration data for diagnostics and excluded held-out products from training and checkpoint selection. All three shared arms assigned equal total sampling mass to each program and preserved the frozen within-program source weights. We also checked the prespecified numerical acceptance criteria. With three training seeds, these checks describe the observed runs and do not establish population-level non-inferiority. Zero program-fixed-coordinate violations are an implementation check required by Proposition C.6, not learned behavior.

B.1.2 COMPARATORS AND MATCHED CONTROLS

The null and cyclic controls tested different aspects of conditioning. Null generation followed by exact post-hoc filtering measured performance without program information. The cyclic arm isolated the reaction-identity label with the remaining program structure retained. A separate train-only finite-component assembler received the exact training components, supported forward chemistry, source-weighted role marginals and the same number of attempts. We evaluated reaction validity and exploration separately because catalogue assembly has zero input-component novelty by construction. None of these comparisons called a precursor-preparation planner, biological oracle or candidate-selection procedure.

For Ugi, we evaluated a common external-baseline protocol with one reaction-space comparator, RGFN, and two generic whole-molecule comparators, unconditional DeFoG and GenMol/SAFE (Koziarski et al., 2024; Qin et al., 2025; Lee et al., 2025). The Synthesis-DAG method of Ou et al. (2024), SynFlowNet and SynCoGen were discussed as related methods and were excluded from the empirical comparison (Cretu et al., 2025; Rekesh et al., 2026). At the version checks used to construct the benchmark, the Synthesis-DAG repository lacked the paper-specific ionizable-lipid implementation, the SynFlowNet action space did not support the qualified three-reactant Ugi transform, and the SynCoGen repository lacked a usable code license. We preserved each method's native reaction operations and included numerical results only from executions using the common training split, attempt budget, seed set and molecular and exact-L1 evaluator. The comparison excludes published scores from other datasets and unsupported aza-Michael or reductive-amination runs. The matched internal suite included unconditioned generation plus post-hoc filtering, an oracle finite-catalogue assembler, a learned inventory selector, two learned component-factorized generators, and targeted interventions on the Transformer. Appendix B.3 reports the three-seed Transformer

mechanism and component-factorization study. Each intervention used the same folds, generation budget and evaluation code as the full Transformer.

B.1.3 METRICS, MOLECULAR IDENTITY AND UNCERTAINTY

Let $Y^{\mathrm{L1}}_{s,a,b} \in \{0,1\}$ indicate verified exact L1 for training seed s, arm a and attempt b, and let $B = 3{,}072$. We report the exact count $C^{\mathrm{L1}}_{s,a} = \sum_{b=1}^{B} Y^{\mathrm{L1}}_{s,a,b}$ and

$$\widehat{p}^{\mathrm{L1}}_{s,a} = C^{\mathrm{L1}}_{s,a}/B. \tag{28}$$

Validity and connected yield replace Y^{L1} with their attempt-level indicators. Verified decomposition coverage is the number of valid graphs with $\widehat{\mathcal{E}}_c(G) \neq \varnothing$ divided by valid graphs; the implementation does not report candidate-search coverage before replay. Exact ambiguity is the fraction of valid graphs with $|\widehat{\mathcal{E}}_c(G)| > 1$. Replay of members of $\widehat{\mathcal{E}}_c(G)$ is an invariant, whereas candidate replay precision is defined in Appendix A.5. The evaluated artifacts do not retain the pre-replay candidate sets, so candidate-search coverage and candidate replay precision were not estimated. Explicitly, with attempt outcomes $G_{s,a,b}$ and the convention that a failed decode makes every graph predicate zero,

$$\widehat{p}^{\mathrm{valid}}_{s,a} = B^{-1} \sum_{b} \mathbf{1}\{\mathrm{Valid}(G_{s,a,b}) = 1\}, \tag{29}$$

$$\widehat{p}^{\mathrm{conn}}_{s,a} = B^{-1} \sum_{b} \mathbf{1}\{\mathrm{Valid}(G_{s,a,b}) = 1, \mathrm{Connected}(G_{s,a,b}) = 1\}, \tag{30}$$

$$\mathrm{CandidateCoverage} = \frac{\sum_b \mathbf{1}\{\mathrm{Valid}(G_b) = 1, \mathcal{D}_c(G_b) \neq \varnothing\}}{\sum_b \mathbf{1}\{\mathrm{Valid}(G_b) = 1\}}, \tag{31}$$

$$\mathrm{ExactAmbiguity} = \frac{\sum_b \mathbf{1}\{\mathrm{Valid}(G_b) = 1, |\widehat{\mathcal{E}}_c(G_b)| > 1\}}{\sum_b \mathbf{1}\{\mathrm{Valid}(G_b) = 1\}}. \tag{32}$$

Each ratio is N/E when its denominator is zero.

Distinct-product yield counts canonical constitutions once. The reported verifier-recovered component-novel yield also counts distinct products and requires an unambiguous recovered input with an out-of-training-catalogue precursor. It does not measure strong catalogue escape (Definition C.5). Held-component recovery counts qualifying attempts, including duplicate products, whose unambiguous input contains a designated held constitution.

For one chemistry c, suppress seed and arm indices. Let $\mathcal{A}_{\mathrm{L1}}$ contain the attempt indices with valid graphs and verified exact L1, and let $\mathcal{A}_{\mathrm{one}} = \{b \in \mathcal{A}_{\mathrm{L1}} : |\widehat{\mathcal{E}}_c(G_b)| = 1\}$. For $b \in \mathcal{A}_{\mathrm{one}}$, write the sole recovered input as $\mathbf{C}_b$, with components $C_{b,j}$ and roles $r_{b,j}$. Let $\mathcal{C}_{\mathrm{train},r}$ and $\mathcal{H}_{\mathrm{held},r}$ be the training and designated held constitution sets for role r, respectively. Define

$$\mathcal{A}_{\mathrm{novel}} = \{b \in \mathcal{A}_{\mathrm{one}} : \exists j,\ C_{b,j} \notin \mathcal{C}_{\mathrm{train},r_{b,j}}\}, \tag{33}$$

$$\mathcal{A}_{\mathrm{held}} = \{b \in \mathcal{A}_{\mathrm{one}} : \exists j,\ C_{b,j} \in \mathcal{H}_{\mathrm{held},r_{b,j}}\}. \tag{34}$$

With can denoting constitutional canonicalization, the yields for $B > 0$ are

$$\mathrm{DistinctL1Yield} = \frac{|\{\mathrm{can}(G_b) : b \in \mathcal{A}_{\mathrm{L1}}\}|}{B}, \tag{35}$$

$$\mathrm{ComponentNovelYield} = \frac{|\{\mathrm{can}(G_b) : b \in \mathcal{A}_{\mathrm{novel}}\}|}{B}, \tag{36}$$

$$\mathrm{HeldComponentYield} = \frac{|\mathcal{A}_{\mathrm{held}}|}{B}. \tag{37}$$

Reported values per 1,000 attempts multiply these ratios by 1,000. Ambiguous decompositions do not contribute to the component-novel or held-component numerators.

For a fixed role, let n_i be the occurrence count of component identity i among unambiguous exact-L1 inputs. When $n_{\mathrm{tot}} = \sum_i n_i > 0$, set $w_i = n_i/n_{\mathrm{tot}}$ and compute

$$N_{\mathrm{eff}} = \exp\left(-\sum_{i:n_i>0} w_i \log w_i\right). \tag{38}$$

This is the exponential of Shannon entropy with natural logarithms. Duplicate products and repeated component occurrences contribute to the counts; N_{eff} is N/E when $n_{\text{tot}} = 0$. Molecular diversity is the mean pairwise ECFP4 Tanimoto distance among distinct valid canonical product constitutions, so molecular duplicates are removed for that metric. For distinct products $g_1, \ldots, g_m$,

$$\text{Diversity} = \frac{2}{m(m-1)} \sum_{1 \le i < j \le m} \left[1 - \text{Tanimoto}\big(\text{ECFP4}(g_i), \text{ECFP4}(g_j)\big)\right], \tag{39}$$

with N/E for $m < 2$. Every other undefined conditional denominator is likewise reported N/E.

Identity is constitutional and stereo-free. Graph and product identity use connected canonical RDKit SMILES with atom maps and stereochemical tags removed; formal charges and bond orders are retained; salts or disconnected products are invalid; and no tautomer normalization is applied. Precursor identity uses the same canonicalization. Precursor-tuple equality is role- and program-step-aware, with repeated inputs compared in their registered step order. Training atom mappings and recovered assembly traces serve different roles. The former establish semantic alignment; the latter record precursor assignments and reaction steps. Neither is part of constitutional identity.

For a paired contrast, $D_s = \widehat{p}_{s,\text{FORGE}} - \widehat{p}_{s,\text{control}}$. We report (D_1, D_2, D_3), $\overline{D}$, and $[\min_s D_s, \max_s D_s]$. These summarize the three observed training seeds and are not 95% confidence intervals. Conditional on one fixed trained model, $C^{\text{L1}}_{s,a}$ has the finite-attempt binomial variation in Proposition C.10; that uncertainty does not substitute for independent retraining. Unless explicitly named primary, architecture, coordinate, guidance, potency and catalogue contrasts are exploratory and no multiplicity-adjusted inferential claim is made.

B.2 Assembly quality and program conditioning

B.2.1 Shared generation across three chemistries

The full shared-model comparison used 3,072 held-out generation attempts per program in each of three independent training seeds (Table 3). Table 1 summarizes the exact-L1 yields. Every verifier-returned trace passed forward replay, as required by the verifier invariant. Mean exact- decomposition ambiguity was 61.0% for repeated aza-Michael addition and 74.7% for repeated reductive amination, and the latter program showed substantial between-seed variation. Table 3 reports molecular validity, connectedness, decomposition coverage, forward-replay precision, abstention, ambiguity, novelty and diversity separately for each program.

The final conditioned model recorded zero fixed-coordinate violations and zero support overflows across 27,648 held-out attempts (three seeds, three programs and 3,072 attempts per program). Exact-assembly yields differed across reaction families despite preservation of all fixed coordinates and graph-support bounds.

Table 3: **Matched shared-model production comparison across reaction programs.** Point estimates report the mean across three independent seeds; the exact-L1 column additionally reports the sample standard deviation. Paired contrasts are summarized by their three seed values, mean and observed range. Conditioned-model counts by seed appear in Table 4. Validity and connectedness use all generation attempts as the denominator. Decomposition coverage, replay precision, abstention and ambiguity retain their prespecified conditional denominators. Novelty is conditioned on valid products or unique returned decompositions, as appropriate. Fraction pairs and exact-L1 yield are percentages; diversity is mean pairwise ECFP4 distance and N_{eff} is the effective generated-component count. Results are never pooled across reaction programs.

	Ugi			Aza-Michael			Reductive amination		
	FORGE	Shared null	Cyclic	FORGE	Shared null	Cyclic	FORGE	Shared null	Cyclic
Valid./Conn.	96.5/96.5	35.9/35.9	93.9/93.9	72.2/72.2	27.4/27.4	71.5/71.5	60.4/60.4	41.9/41.9	59.1/59.1
Decomp./Replay	99.9/100.0	97.0/100.0	99.8/100.0	100.0/100.0	69.5/100.0	100.0/100.0	86.6/100.0	88.4/100.0	85.7/100.0
Abst./Ambig.	0.1/0.0	3.0/0.9	0.2/0.0	0.0/61.0	30.5/29.3	0.0/62.3	13.4/74.7	11.6/76.6	14.3/73.8
Exact-L1/attempt	96.4 ± 2.1	34.8 ± 7.1	93.8 ± 4.2	72.2 ± 1.6	19.2 ± 4.1	71.5 ± 2.2	52.3 ± 3.4	37.0 ± 2.2	50.7 ± 4.9
Whole/Comp. novelty	99.5/96.7	99.9/99.0	99.4/96.9	89.8/85.3	99.6/99.7	90.2/86.3	67.4/97.5	91.5/99.8	66.0/97.0
Diversity/N_{eff}	0.643/419.9	0.680/439.5	0.646/427.9	0.593/118.0	0.707/413.5	0.603/119.4	0.776/173.0	0.761/193.3	0.773/157.8

Table 4: **Exact verified-L1 counts for the final conditioned model.** Each row reports the numerator, fixed attempt denominator and resulting percentage for one independently trained seed. These attempt-level counts quantify conditional Monte Carlo variation; the three training seeds are the independent units for retraining variation. Seed labels 0, 1 and 2 correspond to RNG seeds 20260825, 20260826 and 20260827, respectively.

Program	Seed	Verified L1 count	Attempts	Yield (%)
Ugi	0	2,922	3,072	95.12
Ugi	1	3,037	3,072	98.86
Ugi	2	2,924	3,072	95.18
Aza-Michael	0	2,176	3,072	70.83
Aza-Michael	1	2,203	3,072	71.71
Aza-Michael	2	2,274	3,072	74.02
Reductive amination	0	1,561	3,072	50.81
Reductive amination	1	1,727	3,072	56.22
Reductive amination	2	1,535	3,072	49.97

B.2.2 CONDITIONED, NULL AND CYCLIC CONTROLS

Under the matched protocol in Table 3, the paired conditioned-minus-null differences were 61.6 percentage points for Ugi (three-seed range 57.4 to 67.7), 53.0 for aza-Michael (50.2 to 56.4), and 15.4 for reductive amination (14.5 to 16.7).

The cyclic control retained precursor roles, reaction-core coordinates and program depth but assigned the wrong reaction identity. Relative to correct conditioning, this intervention changed verified exact-L1 yield by 2.6 percentage points for Ugi (three-seed range 0.7 to 5.5), 0.6 for aza-Michael (-2.4 to 3.8), and 1.6 for reductive amination (-0.1 to 2.5). Every returned exact trace passed replay by verifier construction.

We independently retrained the null and cyclic arms under the same study protocol to measure how program information affected learned generation. Correct conditioning improved yield substantially relative to the null arm. Its smaller advantage over cyclic labels did not establish a strong independent contribution from the reaction-identity token.

B.3 DECODER AND ARCHITECTURE ABLATIONS

B.3.1 TERMINAL DECODING AND SOURCE MARGINALS

We evaluated terminal decoding and source marginals in separate fixed-weight comparisons (Table 5).

Table 5: **Decoder and source-marginal ablations of the final model.** Ugi held-out attempts for the 9,143-step shared production model. The two ablations are independent: the decoder rows share one set of trained weights and differ only in terminal decoding, while the source rows share one decoder and differ only in the marginal the corruption starts from. Decoding under the reaction-core saturation policy holds the aldehyde-derived α-carbon at one hydrogen and the isocyanide-derived amide nitrogen at one hydrogen, which the qualified transform fixes; it changes the generated graph under an unchanged verifier. Values are percentages from 3,072 seed-0 held-out attempts per row; they are descriptive and do not measure across-training-seed variation. Exact-L1 coverage is conditioned on valid products.

Source marginal	Terminal decoder	Valid	Exact-L1 coverage	Exact-L1
Program-role	Core saturation	95.21%	99.90%	95.12%
Program-role	Valence topology only	95.31%	78.35%	74.67%
Global	Core saturation	86.95%	99.63%	86.62%

B.3.2 JOINT DENOISING, ROLE ISOLATION AND TRANSFORMER MODULES

The matched-study full Transformer produced 856.9 Ugi and 767.7 repeated aza-Michael exact-L1 products per 1,000 attempts, compared with 56.9 and 396.4 for the parameter-matched role-isolated

FACT model (Table 6). The full Transformer and FACT-matched model each had 5.33 million trainable parameters; FACT-generous had 29.40 million. The higher-capacity FACT control improved to 455.0 for Ugi and 685.7 for repeated aza-Michael addition. Its three-seed means remained below the full Transformer for both programs, although the aza-Michael values were essentially tied in one seed. The direction reversed for repeated reductive amination: the full Transformer produced 318.8 exact-L1 products per 1,000 attempts, whereas FACT-matched and FACT-generous produced 520.3 and 558.1. Thus, joint whole-lipid denoising improved exact assembly for Ugi and repeated aza-Michael addition but was not uniformly superior across reaction programs. Because the prespecified conditional-dependence diagnostic was negative, this architecture contrast does not establish detectable cross-role dependence in the present data.

Table 6: **Transformer architecture and mechanism ablations.** The five Transformer intervention rows change only the named mechanism from the matched-study full model; the two FACT rows generate precursor roles independently at matched or increased capacity. All arms use identical data folds, corruption, context schedule, checkpoint rule, core-saturation terminal decoder and sampling budget. The reference full Transformer is the matched mechanism-study control, not the independently trained production model reported in Table 3 and Appendix B.4. Entries report the mean $\pm$ sample standard deviation across three independent training seeds. Held-out loss is the mean denoising loss at $t = 0.5$ across the three component-disjoint program partitions. Cross-role fidelity is the Frobenius distance to the held-out residualized descriptor matrix, for which lower values indicate closer agreement. Exact-L1 yields use all 3,072 attempts per program and seed as the denominator.

Arm	Held-out loss	Ugi L1/1k	BL L1/1k	LX L1/1k	Held-comp. L1/1k	Cross-role fidelity	Diversity
Input-only program	8.48 ± 0.72	795.2 ± 112.4	776.5 ± 33.2	268.1 ± 56.9	0.00 ± 0.00	1.83 ± 0.15	0.597 ± 0.009
No role loss	7.87 ± 0.12	846.2 ± 98.2	769.6 ± 45.6	289.7 ± 148.1	0.22 ± 0.19	1.92 ± 0.07	0.587 ± 0.011
No core loss	7.85 ± 0.11	833.4 ± 101.7	776.0 ± 37.2	310.1 ± 130.1	0.22 ± 0.19	1.82 ± 0.15	0.584 ± 0.007
No routed adapters	7.81 ± 0.32	791.0 ± 164.9	791.0 ± 49.5	329.9 ± 60.9	0.11 ± 0.19	1.77 ± 0.07	0.587 ± 0.001
No gradient conflict control	7.98 ± 0.20	866.4 ± 84.1	774.1 ± 44.2	324.1 ± 157.4	0.11 ± 0.19	1.92 ± 0.12	0.584 ± 0.008
FACT-matched	7.09 ± 0.24	56.9 ± 24.4	396.4 ± 63.5	520.3 ± 62.2	0.11 ± 0.19	1.31 ± 0.22	0.555 ± 0.014
FACT-generous	8.24 ± 0.31	455.0 ± 191.9	685.7 ± 107.4	558.1 ± 63.6	1.95 ± 0.86	1.12 ± 0.04	0.589 ± 0.005
FORGE	7.88 ± 0.22	856.9 ± 86.7	767.7 ± 35.9	318.8 ± 158.3	0.11 ± 0.19	1.90 ± 0.13	0.585 ± 0.007

Removing layerwise program cross-attention reduced mean exact-L1 yield by 61.6 products per 1,000 attempts for Ugi and 50.7 for repeated reductive amination, while increasing repeated aza-Michael yield by 8.8. Removing the role or core objective and removing routed adapters also produced program-dependent changes. The arm without gradient-conflict control slightly exceeded the full Transformer mean in all three programs. These interventions did not support a uniform benefit from every module. We reported effects separately by chemistry and did not combine them into an architecture score.

B.4 Component novelty, generalization and structural realism

B.4.1 Common Ugi benchmark

We compared FORGE with external generators, matched internal controls and the finite-catalogue oracle under the common Ugi benchmark (Table 2). Every yield is normalized by the number of generation attempts, including attempts that fail before producing a valid molecule. All included methods were run under the common split, attempt budget, seed set and verifier.

The learned finite-inventory selector achieved 1000.0 ± 0.0 verified exact-L1 products per 1,000 attempts, but its finite action space imposed zero verifier-recovered component-novel yield. FORGE produced 963.9 ± 21.4 verified exact-L1 products and 862.6 ± 19.2 distinct verified exact-L1 products per 1,000 attempts. Of these, 839.2 ± 26.5 were component-novel. Among whole-molecule external generators, unconditional DeFoG produced 280.6 ± 26.6 verified exact-L1 products per 1,000 attempts and recovered held components at 15.5 ± 2.4 per 1,000 attempts, compared with 1.4 ± 1.7 for FORGE. GenMol/SAFE produced 627.5 ± 6.2 valid connected outputs but no exact Ugi-L1 product, while RGFN emitted no admissible molecular output under the strict native contract. These results established different performance orderings for reaction consistency, verifier-recovered component novelty and recovery of reserved components. They do not establish product-level strong catalogue escape.

Seed-level results and decomposition outcomes are reported in Tables 7 and 8. All values derive from hash-pinned three-seed results. Compute records are retained in the generated evidence receipt, but missing training-time fields prevent a cross-method compute-matching claim.

Table 7: **Seed-level common Ugi benchmark results.** Each method has a hash-verified three-seed result vector. Table 2 reports the corresponding mean ± sample standard deviation. Training seeds are the independent units; generated molecules are samples from each fixed model.

Method	Seed(s)	Valid/1k	Verified L1/1k	Distinct L1/1k	Comp.-novel/1k	Held-comp. L1/1k	Diversity
RGFN	20260825	0.0	0.0	0.0	0.0	0.0	N/E
RGFN	20260826	0.0	0.0	0.0	0.0	0.0	N/E
RGFN	20260827	0.0	0.0	0.0	0.0	0.0	N/E
DeFoG unconditional	20260825	299.2	251.0	251.0	244.5	13.7	0.711
DeFoG unconditional	20260826	359.7	288.4	288.4	280.6	18.2	0.733
DeFoG unconditional	20260827	380.5	302.4	302.4	291.0	14.6	0.736
GenMol/SAFE	20260825	627.3	0.0	0.0	0.0	0.0	0.883
GenMol/SAFE	20260826	633.8	0.0	0.0	0.0	0.0	0.869
GenMol/SAFE	20260827	621.4	0.0	0.0	0.0	0.0	0.866
Learned inventory selector	20260825	1000.0	1000.0	863.0	0.0	0.0	0.717
Learned inventory selector	20260826	1000.0	1000.0	835.6	0.0	0.0	0.717
Learned inventory selector	20260827	1000.0	1000.0	849.9	0.0	0.0	0.715
Finite catalogue oracle	20260825	1000.0	1000.0	841.5	0.0	0.0	0.719
Finite catalogue oracle	20260826	1000.0	1000.0	843.1	0.0	0.0	0.719
Finite catalogue oracle	20260827	1000.0	1000.0	828.1	0.0	0.0	0.719
Shared-null + post-hoc	20260825	183.9	114.3	109.7	109.7	0.3	0.738
Shared-null + post-hoc	20260826	187.5	103.5	98.3	98.3	0.7	0.763
Shared-null + post-hoc	20260827	212.6	125.3	117.5	117.2	0.3	0.761
FACT-matched	20260825	59.2	49.5	48.8	48.8	0.0	0.676
FACT-matched	20260826	81.7	68.7	66.1	63.2	0.0	0.658
FACT-matched	20260827	41.3	37.8	37.4	37.1	0.0	0.668
FACT-generous	20260825	391.0	161.8	161.1	161.1	0.3	0.782
FACT-generous	20260826	747.1	508.8	488.3	486.7	0.0	0.768
FACT-generous	20260827	363.9	219.4	215.5	215.5	0.0	0.771
FORGE	20260825	952.1	951.2	872.4	852.9	3.3	0.656
FORGE	20260826	990.2	988.6	875.0	856.1	1.0	0.656
FORGE	20260827	952.5	951.8	840.5	808.6	0.0	0.618

Table 8: **Reaction decomposition outcomes under the common Ugi verifier.** Verified decomposition coverage is conditioned on molecular validity. Replay of every tuple in the returned exact set is a verifier invariant, not an estimate of reverse-search precision. Abstention and exact ambiguity retain their prespecified denominators. N/E denotes a conditional metric with no eligible valid product or returned decomposition.

Method	Valid products	Verified coverage	Replay invariant	Abstention	Exact ambiguity
RGFN	0.0 ± 0.0	N/E	N/E	N/E	N/E
DeFoG unconditional	1064.3 ± 129.9	81.2 ± 2.4	100.0 ± 0.0	18.8 ± 2.4	0.0 ± 0.0
GenMol/SAFE	1927.7 ± 19.0	0.0 ± 0.0	N/E	100.0 ± 0.0	0.0 ± 0.0
Learned inventory selector	3072.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Finite catalogue oracle	3072.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Shared-null + post-hoc	598.0 ± 47.9	58.8 ± 3.5	100.0 ± 0.0	41.2 ± 3.5	0.0 ± 0.0
FACT-matched	186.7 ± 62.1	86.3 ± 4.4	100.0 ± 0.0	13.7 ± 4.4	0.0 ± 0.0
FACT-generous	1538.0 ± 656.9	56.6 ± 13.7	100.0 ± 0.0	43.4 ± 13.7	0.0 ± 0.0
FORGE	2964.3 ± 67.3	99.9 ± 0.0	100.0 ± 0.0	0.1 ± 0.0	0.0 ± 0.0

B.4.2 CATALOGUE COMPARISON AND COMPONENT DIVERSITY

At the matched sampling budget, FORGE produced 839.2 verifier-recovered component-novel Ugi products per 1,000 attempts, averaged across the three independent seeds. The corresponding values were 159.3 for aza-Michael addition and 58.7 for reductive amination. The finite-component assembler produced zero by construction because every sampled product uses only components from the training catalogue (Table 9). The catalogue achieved exact-L1 yields of 1000.0, 723.2 and 1000.0 products per 1,000 attempts, compared with 963.9, 721.9 and 523.3 for FORGE. The corresponding raw validities were 1000.0, 723.2 and 1000.0 for the catalogue and 965.0, 722.2 and 604.3 for FORGE. We report these quantities separately because reaction validity and verifier-relative component novelty measure different capabilities. Direct membership in the complete catalogue-reachable product set was not computed, so these results are not reported as strong catalogue escape.

In the distinct-product Ugi role analysis, 19,520 of 26,235 products (74.4%) contained at least one verifier-returned component constitution absent from the training registry. Across all Ugi products included in the novelty analysis, 21,960 of 26,235 distinct products (83.7%) were absent from the full 112,386-product constitutional reference corpus. FORGE effective component counts were 419.9, 118.0 and 173.0, compared with 86.4, 23.4 and 16.9 for the finite catalogue. The Ugi catalogue contained 185 amines, 75 aldehydes and 35 isocyanides and covered 100% of training products but 0% of the calibration and component-disjoint held-out products. Its Cartesian tuple ceilings were 485,625 for Ugi, 774 for aza-Michael and 140 for reductive amination; these count possible component-program inputs, whose forward products may overlap. The catalogue comparison therefore quantified both assembly consistency and the expansion of recovered precursor identities beyond that inventory.

Table 9: **Whole-graph generation versus a finite-component assembler.** Verified exact-L1, distinct exact-L1 and verifier-recovered component-novel yields are reported per 1,000 generation attempts. A component-novel product has exactly one verifier-returned exact-L1 decomposition and contains at least one component constitution absent from the training catalogue. This verifier-relative metric is not strong catalogue escape because complete catalogue-product reachability was not computed. The catalogue has zero input-component novelty by construction, but may achieve higher raw validity or exact-L1 yield. FORGE-minus-catalogue contrasts use the three seeds as descriptive paired units; no three-seed confidence interval is inferred.

	Ugi		Aza-Michael		Reductive amination	
	FORGE	Finite cat.	FORGE	Finite cat.	FORGE	Finite cat.
Valid/1k	965.0	1000.0	722.2	723.2	604.3	1000.0
Verified exact L1/1k	963.9	1000.0	721.9	723.2	523.3	1000.0
Distinct exact L1/1k	862.6	837.6	566.1	147.5	270.6	52.1
Verifier-recovered component-novel/1k	839.2	0.0	159.3	0.0	58.7	0.0
Whole novelty	99.5	16.9	89.8	39.8	67.4	44.8
Component novelty	96.7	0.0	85.3	0.0	97.5	0.0
Diversity/N_{eff}	0.643/419.9	0.719/86.4	0.593/118.0	0.402/23.4	0.776/173.0	0.514/16.9

B.4.3 HELD-COMPONENT RECOVERY

The component-disjoint Ugi fold was never used for training, checkpoint selection or inventory construction. Under the common native-generation assessment, the full Transformer recovered only 13 exact-L1 products containing a designated held component in 9,216 attempts, or 1.4 ± 1.7 products per 1,000 attempts averaged across three seeds. Recovery of specifically reserved components was therefore weak despite the broader component novelty observed above. Novelty accepts any recovered precursor absent from training; held-component recovery requires a match to a designated withheld identity. The results do not establish strong held-component recovery, role-specific held-component coverage or zero-shot generalization to held reaction families.

B.4.4 STRUCTURAL REALISM

Table 10: **Method-blind structural-realism diagnostic.** Every entry is mean $\pm$ sample standard deviation across three training seeds and retains all 3,072 requested attempts in the denominator. FP and descriptor P/C report manifold precision per 1,000 attempts and coverage of the source-study-held-out R0 reference in percent. C2ST is grouped classifier two-sample area under the receiver operating characteristic curve; 0.5 denotes chance discrimination. N_{eff} is the Shannon effective molecule count. N/E denotes a metric that cannot be estimated. This post-hoc structural diagnostic is not a property, activity or synthesis-success score.

Method	Connected/1k	FP P/C	Desc. P/C	C2ST	N_{eff}	Diversity
RGFN	0.0 ± 0.0	$0.0 \pm 0.0/0.00 \pm 0.00$	$0.0 \pm 0.0/0.00 \pm 0.00$	N/E	0.0 ± 0.0	N/E
DeFoG unconditional	346.5 ± 42.3	$0.0 \pm 0.0/0.00 \pm 0.00$	$17.3 \pm 1.4/1.45 \pm 0.06$	0.961 ± 0.007	1064.3 ± 129.9	0.726 ± 0.013
GenMol/SAFE	627.5 ± 6.2	$0.0 \pm 0.0/0.00 \pm 0.00$	$0.0 \pm 0.0/0.00 \pm 0.00$	N/E	3.2 ± 0.1	0.873 ± 0.009
Finite catalogue oracle	1000.0 ± 0.0	$0.2 \pm 0.2/0.05 \pm 0.05$	$54.6 \pm 4.4/2.02 \pm 0.17$	0.971 ± 0.007	2395.5 ± 31.7	0.720 ± 0.000
Learned inventory selector	1000.0 ± 0.0	$0.1 \pm 0.2/0.02 \pm 0.03$	$56.6 \pm 7.2/1.97 \pm 0.46$	0.974 ± 0.002	2443.0 ± 55.8	0.718 ± 0.001
FORGE	965.0 ± 21.9	$0.0 \pm 0.0/0.00 \pm 0.00$	$29.5 \pm 2.6/2.21 \pm 0.03$	0.972 ± 0.004	2489.2 ± 66.8	0.642 ± 0.022

No evaluated method reproduced the full held-out observed-lipid distribution: where estimable, grouped C2ST AUC remained between 0.961 and 0.974, and fingerprint-manifold precision was zero or near zero (Table 10). GenMol/SAFE's high pairwise diversity was paired with an effective count of only 3.2 ± 0.1 molecules, confirming severe frequency collapse. FORGE retained 2489.2 ± 66.8 effective molecules, but only 29.5 ± 2.6 attempts per 1,000 fell within the descriptor manifold. Diversity, effective molecule count and reference-distribution coverage therefore measured different aspects of the generated populations.

B.5 Exploratory lipid synthesis, formulation and *in vivo* imaging

Three lipids generated by the shared three-family model were synthesized by a one-pot, three-component Ugi reaction combining the corresponding amine, aldehyde and isocyanide. This reactant combination follows the ionizable-lipid synthesis strategy described by Chen et al. (2023). Earlier isocyanide-mediated ionizable-lipid synthesis was reported by Miao et al. (2019). FLuc mRNA LNPs contained the FORGE ionizable lipid, DSPC, cholesterol and a PEG-lipid, with a reported amine-to-phosphate ratio of $5.67 : 1$. Mice received $1\ \mathrm{mg}$ FLuc mRNA per kg body weight. D-luciferin was administered at 4 hours, followed by bioluminescence imaging. The four reported ROI values are reproduced in Figure 2.

The quantitative display reproduces the supplied values without inferring replicate counts, calculating group means or estimating uncertainty. No significance test was applied. The three FORGE formulations are distinct designs and are not pooled as biological replicates.

B.6 Potency-guidance diagnostic

We evaluated biological guidance only in the bounded HeLa messenger-transfection-potential (mTP) diagnostic. A group-disjoint preterminal classifier detected weak morphology-level signal, with ROC AUC 0.623 (95% interval 0.565 to 0.679) and a 1.177-fold observed top-quartile gain. We then compared a frozen applicability-enriched morphology proposal with a nested potency-enriched proposal using 3,072 generator calls per arm. Table 11 reports the matched terminal outcome.

Table 11: **Predicted HeLa messenger-transfection potential under matched generation budgets.** The support-enriched arm used the selected applicability proposal. The nested arm reallocated morphology mass using the frozen weak potency signal. Conservative-high counts are computational predictions, not transfection measurements. The comparison uses the same generator-call and oracle budgets.

Arm	Attempts	Verified L1	Eligible before budget	Oracle budget	Unique conservative-high
Support-enriched	3,072	2955	90	38	27
Nested potency proposal	3,072	2962	82	38	23

The nested-minus-support difference was -1.30 conservative-high products per 1,000 generator calls (95% interval -2.28 to 0.98). The comparison failed the prespecified advancement criterion. These computations evaluate predictor behavior within a declared applicability domain; they do not establish improved biological activity, delivery or experimental potency.

C THEORETICAL ANALYSIS AND PROOFS

The results establish separate properties of the construction. Bounded completeness concerns the state representation. Catalogue reachability concerns exact precursor inputs. The population argument connects a reference denoising loss to an ideal continuous-time law. The sampling-budget result characterizes exact-L1 acceptance counts. None establishes experimental synthesis success or positive learned probability for every representable graph.

C.1 GRAPH STATES AND PROGRAM COMPATIBILITY

The graph space $\mathfrak{G}_{N,K_{\mathrm{cl}}}$ is defined in Section 2.3, with finite atom and bond alphabets and integers $N \geq 1$, $K_{\mathrm{cl}} \geq 0$. Molecular validity is an additional predicate on that space. The raw space $\overline{\mathfrak{X}}_{N,K_{\mathrm{cl}}}$ contains padded categorical states.

Definition C.1 (Well-formed sparse state)**.** A state belongs to $\mathfrak{X}_{N,K_{\mathrm{cl}}}$ when, for some $1 \leq n \leq N$, its active atoms are exactly positions $1, \ldots, n$ with labels in Σ_{atom}; position 1 is the sole root with null parent and parent-bond states; and every $2 \leq i \leq n$ has an earlier parent $1 \leq \pi_i < i$ and a bond label in Σ_{bond}. Each active closure has endpoints $1 \leq e^L < e^R \leq n$, a bond label in Σ_{bond}, and an edge distinct from every tree edge and other closure. Active closures precede unused slots and are sorted by endpoint pair. All unused atom, pointer and bond coordinates carry their designated null values.

The decoder reconstructs the labeled edges of a well-formed state and returns $\perp_{\mathrm{dec}}$ otherwise. It is therefore a total map into $\mathfrak{G}_{N,K_{\mathrm{cl}}} \cup \{\perp_{\mathrm{dec}}\}$.

Lemma C.2 (Tree residual count)**.** *For a finite nonempty connected graph $G = (V, E)$ and any spanning tree T, $|E \setminus E(T)| = |E| - |V| + 1 = \beta(G)$.*

Proof. The tree contains $|V| - 1$ edges, all in E. Subtracting this count from $|E|$ gives the result, including the single-vertex case. □

Theorem C.3 (Bounded completeness of the sparse representation)**.** *For the spaces in Definition C.1, the restriction* $\mathrm{Dec} : \mathfrak{X}_{N,K_{\mathrm{cl}}} \to \mathfrak{G}_{N,K_{\mathrm{cl}}}$ *is well defined and surjective. Thus every well-formed state decodes to a graph in the declared support, and every graph in that support has an encoding that decodes to its constitutional isomorphism class.*

Proof. We first reconstruct a graph from a state, then construct a state for an arbitrary graph. In a well-formed state with n active atoms, every parent pointer strictly decreases its index. Following parents must reach position 1, so the $n - 1$ parent edges form a connected tree. Adding $k \leq K_{\mathrm{cl}}$ distinct, non-loop closure edges preserves connectedness and gives cycle rank $(n-1+k)-n+1 = k$. The label and padding conditions ensure the decoded graph belongs to $\mathfrak{G}_{N,K_{\mathrm{cl}}}$.

Conversely, choose a representative of any $G \in \mathfrak{G}_{N,K_{\mathrm{cl}}}$ and a rooted spanning tree. Enumerate vertices in a traversal that places every parent before its children. Store their labels, parent indices and tree-bond labels. Lemma C.2 leaves exactly $\beta(G) \leq K_{\mathrm{cl}}$ residual edges; store each with increasing endpoint indices and sort these triples. Pad all unused coordinates with null states. This state satisfies Definition C.1, and the vertex enumeration is a label-preserving isomorphism between its decoded graph and G. □

This result concerns the unrestricted representation. For program-specific alphabets $\mathcal{X}_i(P)$ and fixed values $x_{P,i}$, define

$$\Omega_P = \{x \in \overline{\mathfrak{X}}_{N,K_{\mathrm{cl}}} : x_i \in \mathcal{X}_i(P)\ \forall i, \quad x_i = x_{P,i}\ \forall i \in \mathcal{F}(P)\}. \tag{40}$$

Its nonempty well-formed intersection defines $\mathcal{P}_{\mathrm{adm}} = \{P : \mathfrak{X}_{N,K_{\mathrm{cl}}} \cap \Omega_P \neq \varnothing\}$. Nonemptiness alone establishes neither chemical validity nor satisfaction of every morphology constraint. Corpus admission additionally checks the annotated encoding, program alignment and exact decode round trip. Representational completeness makes no claim that every graph is encodable under every P.

C.2 Exact assembly and catalogue reachability

Definition A.1 distinguishes the complete input preimage $\mathcal{E}_c^\star(G)$ from recovered input–trace pairs $\widehat{\mathcal{E}}_c(G)$. Fix a registered chemistry c and role-specific catalogues $\mathcal{C} = (\mathcal{C}_r)_r$. Let $\mathfrak{C}_c(\mathcal{C}) \subseteq \mathfrak{C}_c$ contain the admissible precursor inputs whose every role-assigned component belongs to its catalogue. The reachable product set is

$$\mathcal{G}_{\mathrm{cat}}(c, \mathcal{C}) = \bigcup_{\mathbf{C} \in \mathfrak{C}_c(\mathcal{C})} \mathcal{F}_c(\mathbf{C}). \tag{41}$$

This definition includes all registered outcomes of the admissible inputs, not only products found by a bounded search.

Proposition C.4 (Finite-component reachability)**.** *If $\mathfrak{C}_c(\mathcal{C})$ and each $\mathcal{F}_c(\mathbf{C})$ are finite, then*

$$|\mathcal{G}_{\mathrm{cat}}(c, \mathcal{C})| \le \sum_{\mathbf{C} \in \mathfrak{C}_c(\mathcal{C})} |\mathcal{F}_c(\mathbf{C})| \le m_{\max,c}(\mathcal{C})\, |\mathfrak{C}_c(\mathcal{C})|, \tag{42}$$

where $m_{\max,c}(\mathcal{C})$ is the largest outcome count, or zero for an empty input set. For every $G \in \mathfrak{G}_{N,K_{\mathrm{cl}}}$,

$$G \in \mathcal{G}_{\mathrm{cat}}(c, \mathcal{C}) \quad \Longleftrightarrow \quad \mathcal{E}_c^\star(G) \cap \mathfrak{C}_c(\mathcal{C}) \neq \varnothing. \tag{43}$$

Proof. The cardinality of a finite union is at most the sum of its member-set cardinalities; bounding each summand by its maximum proves both inequalities. If there are no inputs, all three quantities are zero. Membership in the union means that some catalogue input has G as a forward product, which is exactly the stated intersection condition. □

Definition C.5 (Strong catalogue escape)**.** An exactly assemblable graph strongly escapes $\mathcal{C}$ under c when every exact input uses at least one out-of-catalogue component. Its indicator is

$$\mathrm{Escape}^\star(G; c, \mathcal{C}) = \mathbf{1}[\mathcal{E}_c^\star(G) \neq \varnothing, \quad \mathcal{E}_c^\star(G) \cap \mathfrak{C}_c(\mathcal{C}) = \varnothing]\,. \tag{44}$$

A catalogue assembler whose outputs lie in $\mathcal{G}_{\mathrm{cat}}(c, \mathcal{C})$ therefore has zero strong-escape yield. For a verified exact-L1 product, strong escape is equivalent to $G \notin \mathcal{G}_{\mathrm{cat}}(c, \mathcal{C})$. This can be checked against a complete canonical product set or by exhaustive exclusion of all catalogue-only decompositions. A bounded reverse search cannot establish that exclusion from one novel witness.

The reported experiments did not construct a complete common-reference reachable product set. Their component-novelty metric concerns the selected verified input. Whole-graph generation removes catalogue membership from the generative representation; that architectural property alone proves neither strong escape nor positive probability of any particular out-of-catalogue graph.

C.3 Conditional flow and ideal population recovery

We prove Theorem 3.1 for the reference objective in Equation 7. Let $X_\star = \mathrm{Enc}_P(G, \alpha)$ denote a clean training endpoint, and distinguish it from the time-indexed process X_t. The training-path measure is

$$\nu(dG, d\alpha, dP, dt, dx) = \mu_{\mathrm{bal}}(dG, d\alpha, dP)\, \mathcal{U}_{(0,1)}(dt)\, q_{t|1}^P(dx \mid \mathrm{Enc}_P(G, \alpha)). \tag{45}$$

For fixed P, the encoded endpoint law q_1^P induces

$$q_t^P(x) = \sum_{x_1} q_1^P(x_1) q_{t|1}^P(x \mid x_1), \qquad \rho_i(a \mid x, t, P) = \Pr_\nu[X_{\star,i} = a \mid X_t = x, t, P]. \tag{46}$$

Full source support on the finite allowed alphabets makes $q_t^P(x) > 0$ for every $x \in \Omega_P$ and $t < 1$. At $t = 0$ the path is q_0^P; at $t = 1$ it is q_1^P.

Conditional and marginal rates. For $i \in \mathcal{V}(P)$ and $a, u, v \in \mathcal{X}_i(P)$, write $s^P_{t,i}(u \mid a) = (1-t)q^P_{0,i}(u) + t\mathbf{1}\{u = a\}$ and $\dot{s}^P_i(u \mid a) = \mathbf{1}\{u = a\} - q^P_{0,i}(u)$. The conditional rate used in the ideal analysis is the R^* construction of Qin et al. (2025), restricted to the allowed alphabet:

$$R^*_{t,i}(u, v \mid a, P) = \frac{[\dot{s}^P_i(v \mid a) - \dot{s}^P_i(u \mid a)]_+}{|\mathcal{X}_i(P)|\, s^P_{t,i}(u \mid a)}, \qquad u \neq v, \quad 0 \leq t < 1. \tag{47}$$

Diagonal entries are the negative row sums. Denominators are positive for $t < 1$; no value at $t = 1$ is needed. Fixed coordinates have zero rates. For $x \neq y$ in Ω_P, the joint conditional and predicted rates are

$$R^*_t(x, y \mid x_1, P) = \sum_{i \in \mathcal{V}(P)} \mathbf{1}\{y_{-i} = x_{-i}\} R^*_{t,i}(x_i, y_i \mid x_{1,i}, P), \tag{48}$$

$$R^\theta_t(x, y \mid P) = \sum_{i \in \mathcal{V}(P)} \mathbf{1}\{y_{-i} = x_{-i}\} R^\theta_{t,i}(x_i, y_i \mid x, P). \tag{49}$$

Here x_{-i} omits coordinate i, and Equation 6 defines the predicted coordinate rate. Diagonal joint rates again make each row sum to zero. States differing at two or more coordinates have zero instantaneous transition rate.

Proposition C.6 (Program-fixed invariance)**.** *Fix $P \in \mathcal{P}_{\mathrm{adm}}$ and an endpoint in Ω_P. Corruption by Equation 5 stays in Ω_P. The ideal CTMC started from q^P_0 stays there for $t < 1$ whenever its coordinate predictions are probability distributions on the allowed alphabets. Numerical sampling in Equation 9, started from q^P_0 and using the allowed coordinate alphabets, satisfies* $\Pr[Z_k \in \Omega_P \text{ for all } 0 \leq k \leq L_{\mathrm{samp}}] = 1$.

Proof. The corruption law has zero mass outside Ω_P, and the ideal generator has no transition to its complement. Each numerical coordinate draw stays in its allowed alphabet and Π_P restores the prescribed fixed values. Induction on k proves the discrete statement. These arguments do not require an accurate denoiser and do not imply molecular validity. □

Proposition C.7 (Posterior minimizer of the reference loss)**.** *Under the reference training measure ν, assume unrestricted measurable coordinate predictors and positive weights $w_i(P)$ for each variable coordinate. Every population minimizer of Equation 7 satisfies $d_{\theta,i}(a \mid x, t, P) = \rho_i(a \mid x, t, P)$ for all $a \in \mathcal{X}_i(P)$, ν-almost surely on contexts with $i \in \mathcal{V}(P)$.*

Proof. Condition on (X_t, t, P) and a variable coordinate. Since $w_i(P)$ is positive and does not depend on its clean target after this conditioning, its conditional loss is

$$w_i(P)\,[H(\rho_i) + D_{\mathrm{KL}}(\rho_i \| d_{\theta,i})]\,. \tag{50}$$

The entropy is independent of the predictor, and the divergence is nonnegative with equality only at $d_{\theta,i} = \rho_i$. An unrestricted class can attain all these pointwise minima together. A population minimizer must attain them almost surely, since a positive-measure excess would increase the integrated loss. Fixed coordinates contribute no term. □

Corollary C.8 (Population recovery of the local rate)**.** *For the reference population minimizer in Proposition C.7, $R^{\theta^\star}_t(x, y \mid P)$ equals the conditional expectation of $R^*_t(x, y \mid X_\star, P)$ given $(X_t = x, t, P)$, almost surely under the path measure.*

Proof. Each local conditional rate depends on the endpoint only through its coordinate $X_{\star,i}$. Substituting the posterior marginal into Equation 6 therefore computes the required conditional expectation. Summing over coordinates gives the joint rate. □

Proof of Theorem 3.1. We verify the forward equation first for one coordinate, then for the conditional product path, and finally after averaging over endpoints. Fix P, x_1 and i; suppress these indices in s_t and $\dot{s}$. For each v and $t < 1$, the net flux of Equation 47 is

$$\sum_u s_t(u) R_t^*(u, v) = \frac{1}{|\mathcal{X}_i(P)|} \sum_u \big([\dot{s}(v) - \dot{s}(u)]_+ - [\dot{s}(u) - \dot{s}(v)]_+ \big) \tag{51}$$

$$= \dot{s}(v) - \frac{1}{|\mathcal{X}_i(P)|} \sum_u \dot{s}(u) = \dot{s}(v). \tag{52}$$

The last equality uses $\sum_u \dot{s}(u) = 0$. The product rule then gives the conditional joint forward equation because the generator changes one coordinate at a time. Averaging over q_1^P and using Bayes' rule yields

$$\partial_t q_t^P(y) = \sum_{x, x_1} q_1^P(x_1) q_{t|1}^P(x \mid x_1) R_t^*(x, y \mid x_1, P) \tag{53}$$

$$= \sum_x q_t^P(x)\, \mathbb{E}[R_t^*(x, y \mid X_\star, P) \mid X_t = x, t, P]. \tag{54}$$

Corollary C.8 identifies the last conditional expectation with the learned ideal rate for almost every t. On each interval $[0, 1 - \eta]$, $0 < \eta < 1$, source positivity bounds all denominators away from zero. The rates are bounded measurable functions on a finite state space, so the forward equation has a unique solution for initial law q_0^P. Since q_t^P is positive throughout Ω_P, almost-sure equality under the path measure implies rate equality at every state for Lebesgue-almost every t. Changes on the exceptional times do not change the integral forward equation. The exactly simulated process therefore has law q_t^P.

The finite product path converges to the endpoint point mass as $t \uparrow 1$. Summing over its finite endpoint support gives $q_t^P \to q_1^P$ in total variation. This establishes a limit of marginal laws; a pathwise value at the singular endpoint is not required. Exact decode round trip maps each endpoint to its source constitution, proving $\mathrm{Dec}_\# q_1^P = \mu_G^P$. The assumed endpoint validity and verified L1 give the probability-one conclusion. If there are no variable coordinates, the compatible state space is a singleton and the same conclusions hold for the constant process. □

Corollary C.9 (Decoder contraction and failure probability)**.** *Let q_1^P satisfy the endpoint assumptions of Theorem 3.1. Let $\widetilde{q}^P$ be any law on $\overline{\mathfrak{X}}_{N,K_{\mathrm{cl}}} \cup \{\perp_{\mathrm{term}}\}$, with q_1^P extended by zero mass at $\perp_{\mathrm{term}}$ and $\mathrm{Dec}(\perp_{\mathrm{term}}) = \perp_{\mathrm{dec}}$. Define $\mathcal{B}_P$ as states whose decode fails, is molecularly invalid or fails verified L1. Then*

$$\| \mathrm{Dec}_\# \widetilde{q}^P - \mu_G^P \|_{\mathrm{TV}} \le \| \widetilde{q}^P - q_1^P \|_{\mathrm{TV}}, \qquad \widetilde{q}^P(\mathcal{B}_P) \le \| \widetilde{q}^P - q_1^P \|_{\mathrm{TV}}, \tag{55}$$

where $\|p - q\|_{\mathrm{TV}} = \frac{1}{2} \sum_x |p(x) - q(x)|$.

Proof. Every decoder-output event has a preimage in the raw space, so its probability difference is bounded by the raw-state total variation. Taking the supremum over output events proves the first bound. The endpoint assumptions give $q_1^P(\mathcal{B}_P) = 0$; applying the same event bound to $\mathcal{B}_P$ proves the second. □

The corollary can be applied to the law *after* numerical sampling and terminal readout. It does not estimate that law's distance from q_1^P. In particular, error before the learned terminal readout cannot be substituted without controlling the readout's effect.

C.4 Accepted counts under a fixed sampling budget

Fix a trained model, requested chemistry c and its program-sampling policy. An attempt outcome Y contains its sampled program P and decoded graph or typed failure. Let Q be the common law of these outcomes and define

$$A(Y) = \mathbf{1}\{G \text{ is valid and } \widehat{\mathrm{L1}}(G, c) = 1\}, \qquad z_Q = \mathbb{E}_Q[A(Y)], \tag{56}$$

with $A(Y) = 0$ for failures. This definition permits a fresh morphology-specific P on every attempt. A comparison with a single fixed P is the corresponding special case.

Proposition C.10 (Exact law of matched-budget post-hoc filtering)**.** *For an integer $B \geq 0$ and independent attempts with common law Q, the accepted-attempt count $N_B = \sum_{b=1}^{B} A(Y_b)$ is binomial with parameters (B, z_Q). For all $z_Q \in [0, 1]$,*

$$\mathbb{E}[N_B] = Bz_Q, \qquad \mathrm{Var}(N_B) = Bz_Q(1 - z_Q), \qquad \Pr[N_B = 0] = (1 - z_Q)^B. \tag{57}$$

For $B = 0$ the last probability is one, including when $z_Q = 1$. If $z_Q > 0$, the conditional law of an accepted attempt is $Q_{\mathrm{post}}(y) = Q(y)A(y)/z_Q$, and the number T_m of attempts needed for $m \geq 1$ acceptances satisfies $\mathbb{E}[T_m] = m/z_Q$. For two proposal laws under the same attempt budget, $\mathbb{E}[N_B^{(1)} - N_B^{(0)}] = B(z_{Q_1} - z_{Q_0})$.

Proof. Each acceptance indicator is Bernoulli with parameter z_Q, and independence makes their sum binomial. Its expectation and variance give the first two identities; all B attempts fail with probability $(1 - z_Q)^B$. For $z_Q > 0$, conditioning on acceptance gives the normalized law by Bayes' rule. The waiting times between acceptances are independent geometric variables on $\{1, 2, \ldots\}$ with mean $1/z_Q$; summing m of them proves the waiting-time formula. Subtracting the expected counts proves the comparison identity, without requiring independence between the two arms. □

If $z_Q = 0$, accepted outcomes have no conditional law and $T_m = \infty$ almost surely for $m \geq 1$. The proposition counts accepted attempts, including duplicates. It does not establish that program conditioning increases acceptance; that is an empirical comparison. If requested programs are fixed in unequal strata, independent acceptance indicators generally have different probabilities and their sum is Poisson-binomial. Retraining variation is separate from both sampling laws.

D REACTION SCHEMES AND GENERATED EXAMPLES

The assembly schemes depict the chemistry introduced in Section 2.2. The generated examples pair molecular structures with precursor origins and exact assembly traces. Their evidentiary scope is structural and computational.

D.1 ASSEMBLY SCHEMES

The precursor roles and assembly patterns are summarized in Table 12.

Table 12: **Assembly chemistry underlying the three reaction programs.** The named roles specify reactive functionality without selecting a precursor identity. Repeated programs encode the successive additions admitted by the corresponding registered transform.

Reaction family	Role-assigned precursors	Assembly pattern
Ugi 3-CR	Amine, aldehyde and isocyanide	Forms an α-amino amide combining three precursor contributions (Xu et al., 2024).
Repeated aza-Michael addition	Amine head and acrylate-bearing hydrophobic components	Amine addition to acrylate groups forms C–N bonds; successive additions attach hydrophobic components at available amine sites (Li et al., 2023).
Repeated reductive amination	Amine head and aldehyde-bearing hydrophobic components	Condensation followed by reduction forms C–N linkages, attaching hydrophobic components through successive transformations (Xue et al., 2024).

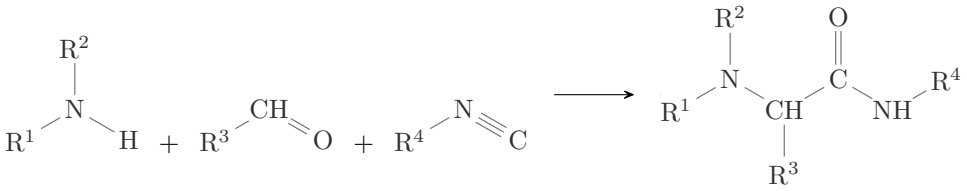
The three registered assembly patterns use skeletal notation and explicit hydrogens to mark substitution constraints at the reaction core (Figure 3). Precursor-origin boundaries can differ from the functional headgroup, linker and tail regions introduced in Section 2.1.

D.1.1 REACTION TRANSFORMS AND EXACT ASSEMBLY

Each assembly chemistry defines a forward transformation that maps role-assigned precursor molecules to the product graphs permitted by that reaction. The corresponding reverse transformation decomposes a complete lipid into the precursor assignments that could have produced it. To verify a decomposition, we computationally replay the reaction on the recovered precursors. The product passes the level-one (L1) assembly check only if the reconstruction matches the original lipid exactly.

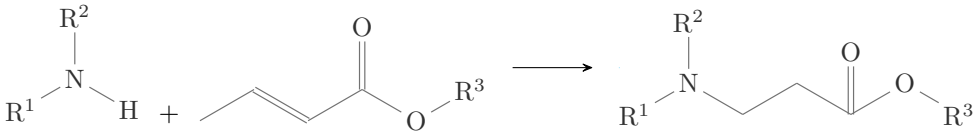
Ugi three-component reaction

amine head, aldehyde and isocyanide combine in one step; no carboxylic-acid reactant component



Repeated aza-Michael addition

the head N–H adds across the acrylate double bond; the ester is untouched



Repeated reductive amination

one new carbon–nitrogen bond; the carbonyl oxygen leaves as water

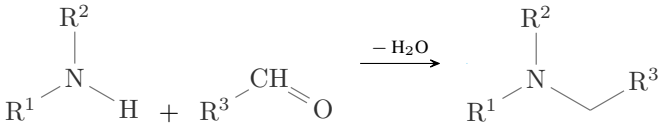


Figure 3: **The three assembly chemistries.** Monochrome skeletal drawings show the reactants and products of each registered assembly. Hydrogens are drawn only on the atoms whose substitution decides whether a product can be decomposed back into precursors: the head N–H, and the α-carbon C–H that the Ugi reaction takes from the aldehyde. R^i stand for the source-linked components of the frozen library, which are larger than the fragments drawn here. The Ugi variant has no carboxylic-acid reactant component.

D.2 GENERATED-STRUCTURE ATLAS

Table 13 presents eight curated display examples from the seed-0 program-projection run. Exact-L1 products were screened by their role-resolved SMILES for a compact tertiary-amine head, an ester-bearing aldehyde-derived body and tail, and a hydrophobic isocyanide-derived tail. This display curation was not used in any reported metric or in an activity or experimental claim. Each row includes the canonical constitutional SMILES to make the displayed identity explicit.

Table 13: **FORGE generated-structure atlas.** Eight example lipids with canonical SMILES (left) and their amine, aldehyde and isocyanide building blocks (right). Each set reconstructs its lipid exactly under the registered Ugi 3-CR transform. This verifies computational L1 assembly, not experimental synthesis.

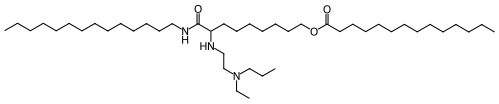
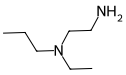
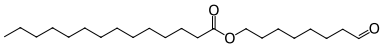
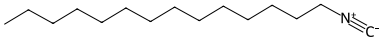
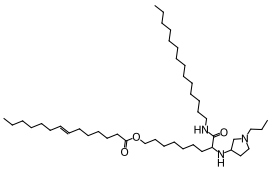
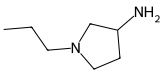
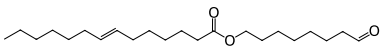
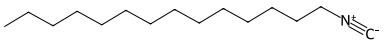
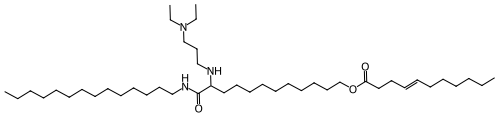
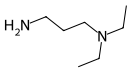
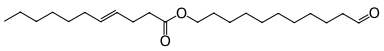
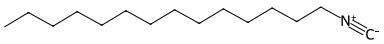
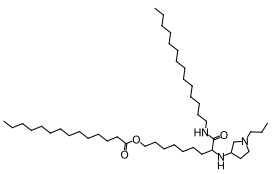
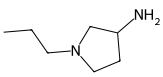
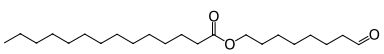
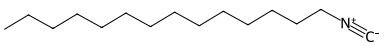
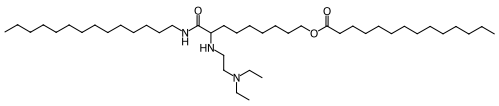
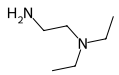
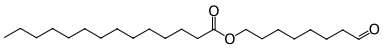
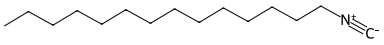
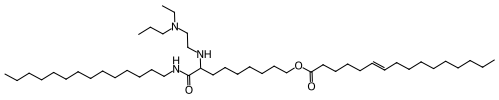
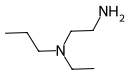
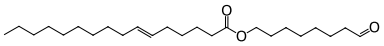
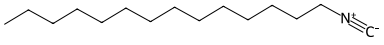
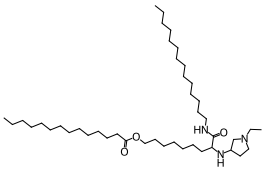
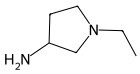
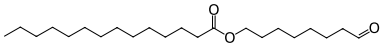
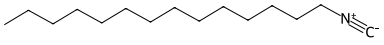
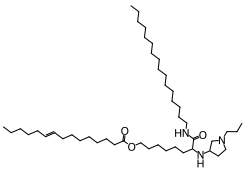
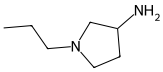
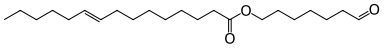
	GENERATED LIPID AND CANONICAL SMILES	EXACT L1 BUILDING BLOCKS
1	 CCCCCCCCCCCCCCNC(=O)C(CCCCCC COC(=O)CCCCCCCCCCCC)NCCN(CC)CCC	Amine  Aldehyde  Isocyanide 
2	 CCCCCCC=CCCCCCC(=O)OCCCCCCCC (NC1CCN(CCC)C1)C(=O)NCCCCCCCCCCCCCC	Amine  Aldehyde  Isocyanide 
3	 CCCCCCC=CCCC(=O)OCCCCCCCCCC (NCCCN(CC)CC)C(=O)NCCCCCCCCCCCCCC	Amine  Aldehyde  Isocyanide 
4	 CCCCCCCCCCCCCCNC(=O)C(CCCCCC COC(=O)CCCCCCCCCCCC)NC1CCN(CCC)C1	Amine  Aldehyde  Isocyanide 

Table 13, continued

	GENERATED LIPID AND CANONICAL SMILES	EXACT L1 BUILDING BLOCKS
5	 CCCCCCCCCCCCCCNC(=O)C(CCCCCC COC(=O)CCCCCCCCCCCCC)NCCN(CC)CC	Amine  Aldehyde  Isocyanide 
6	 CCCCCCCCCC=CCCCCC(=O)OCCCCCC CC(NCCN(CC)CCC)C(=O)NCCCCCCCCCCCCCC	Amine  Aldehyde  Isocyanide 
7	 CCCCCCCCCCCCCCNC(=O)C(CCCCCC COC(=O)CCCCCCCCCCCC)NC1CCN(CC)C1	Amine  Aldehyde  Isocyanide 
8	 CCCCCC=CCCCCCCCC(=O)OCCCCCCC (NC1CCN(CCC)C1)C(=O)NCCCCCCCCCCCCCCCC	Amine  Aldehyde  Isocyanide 