LEARNING SUBSTRUCTURE REGULARITIES FROM WHOLE-GRAPH SEQUENCES

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive Transformers trained on whole graphs can reproduce substructure-level regularities without explicit substructure supervision. We investigate what this agreement reveals about substructure generalization, using PCQM4Mv2 as the main testbed. On PCQM4Mv2, under pure ancestral sampling, a Transformer trained on DFS-code serializations reproduces the support ranking of frequent training substructures, even after exact training-string matches are excluded. Finite-context n-gram models also achieve substantial aggregate agreement, and on five small TUDataset benchmarks agreement coincides with replay, so this observation alone does not identify how substructures are generated. We therefore remove every training graph containing a specified target and retrain Transformers on an alternative serialization of node and edge additions. Under unconditional sampling, models generate structurally valid graphs containing withheld cyclic and acyclic targets across all three training seeds. The rare outputs containing a withheld carbon triangle are not labeled-isomorphic to any graph in the original, unfiltered training set and cannot be produced by generators restricted to recombining retained training fragments only through bridge connections. These results establish both support reproduction and the generation of specific structures absent from training, without identifying an internal mechanism or a general construction rule. Combining support evaluation, sequence baselines, and controlled target removal provides an empirical framework for characterizing observable substructure generalization and for future investigation of its underlying mechanisms.

1 INTRODUCTION

Autoregressive models for graph generation typically operate on serialized token representations of graphs, such as DFS codes (Yan & Han, 2002; Goyal et al., 2020), node/edge-addition sequences (DGMG; Li et al., 2018), SMILES strings (Weininger, 1988; Bagal et al., 2022; Flam-Shepherd et al., 2022), or flattened graphs (Chen et al., 2025). These models are trained on whole-graph sequences; no substructure is an explicit training target. Yet generated graph sets often reproduce substructure frequencies observed in the training data. Our research question in this paper is: What substructure-level regularities do autoregressive models learn when trained only on whole graphs, without explicit substructure supervision?

One of the most direct statistics for characterizing substructures is their *frequency*. However, matching substructure frequencies does not by itself show how a generator produces substructures. Such agreement can arise because the generated distribution is close to the data distribution, and even training-set replay reproduces these statistics. More generally, substantial agreement can persist even when part of the generated distribution is unrelated to the data. We formalize these observations in Section 3.1 and use them to motivate a more direct test of substructure generalization.

We therefore study two observable phenomena: (i) whether generated graphs reproduce the *support* of substructures in the training distribution, i.e., the probability that a graph contains a given substructure, and (ii) whether models generate specified substructures that have been completely removed from training. We reserve *substructure generalization* for the second phenomenon.

We evaluate these phenomena on PCQM4Mv2 (Hu et al., 2021), with auxiliary support analyses on five TUDataset benchmarks (Morris et al., 2020). For support reproduction (Section 4), a LLaMA-

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.

style autoregressive decoder (Vaswani et al., 2017; Touvron et al., 2023) trained on DFS codes reproduces frequent-substructure supports under pure ancestral sampling about as closely as same-size samples of real molecules, including after excluding outputs that copy a training string. However, a non-neural baseline that recombines training fragments reaches comparable agreement, and finite-context n-gram models over DFS tokens reproduce much of the support ranking. Thus, support agreement alone does not identify how the model generates substructures.

We then perform a controlled target-removal experiment (Section 5). For each target substructure, we remove every training graph containing that target and retrain from scratch on node/edge-addition sequences (DGMG-style). This tests withheld-target generation in a second representation, without establishing serialization invariance. We compare against both full-data training and matched-removal controls that remove the same number of length-matched, target-free graphs. Despite never observing the target during training, the models generate structurally valid graphs containing C_3, a withheld triangle of three carbon atoms joined by single bonds. Such generations are rare but occur across all three training seeds. None of these outputs is labeled-isomorphic to any graph in the original, unfiltered training set.

This target is particularly informative because a generator restricted to recombining target-free training fragments only across bridges (bonds that lie on no cycle) cannot create C_3. The observed C_3 generations therefore fall outside this bridge-only explanation. A second withheld target, P_6, is a six-carbon path whose five single bonds are all bridges of the host graph. It remains constructible by bridge-only recombination and is also generated after removal, but substantially less often than under matched-removal controls. Exact-count k-gram models trained on the same withheld data also emit C_3, but removal suppresses their emission much less strongly than it suppresses Transformer emission. Together, these experiments separate support reproduction, which replay and simple baselines also achieve, from strongly suppressed yet non-zero generation of a specified unseen structure beyond bridge-only recombination; they do not identify the generating mechanism (Section 5.3).

Our study provides an empirical framework for characterizing substructure behavior in autoregressive graph generators:

- a support-based evaluation with real-data and non-neural baselines, separating invalid generation from substructure-support error (Sections 3 and 4);
- a controlled target-removal protocol with matched controls and novelty checks, together with a graph-theoretic constraint on what bridge-only recombination can generate (Section 5); and
- a diagnostic analysis showing that held-out likelihood does not uniquely determine substructure-support error across the tested generators, with discrepancies associated with generated graph size (Section 6).

2 Related Work

Graph generation and serialized graphs. Autoregressive generators grow graphs node by node or block by block (You et al., 2018; Li et al., 2018; Liao et al., 2019); diffusion models denoise nodes and edges (Jo et al., 2022; Vignac et al., 2023); fragment-based generators build molecules from an explicit substructure vocabulary (Jin et al., 2018; 2020; Yamada & Sugiyama, 2023), which our models lack. Closest to ours are serialized-graph models: GraphGen trains an LSTM on minimum DFS codes (Goyal et al., 2020), molecular language models learn from strings (Bagal et al., 2022; Flam-Shepherd et al., 2022), and AutoGraph trains a decoder-only Transformer on flattened graphs whose prefixes encode induced subgraphs, supporting substructure-conditioned generation (Chen et al., 2025). We propose no generator; we ask what such a model acquires from whole graphs.

Evaluation and memorization. Graph-generation metrics have known pitfalls (O'Bray et al., 2022; Thompson et al., 2022), a trivial model that perturbs training molecules scores near-perfectly on several GuacaMol distribution-learning metrics (Brown et al., 2019; Renz et al., 2019), and language models can reproduce training sequences verbatim (Carlini et al., 2021; 2023). These findings motivate our controls: real-data resamples, bridge-only fragment-recombination and n-gram baselines, and two novelty notions (cf. Polykovskiy et al., 2020; Section 3.3).

Held-out combinations. SCAN, CFQ, COGS and PCFG SET test combinations of known parts that never co-occur in training (Lake & Baroni, 2018; Keysers et al., 2020; Kim & Linzen, 2020; Hupkes et al., 2020). Target removal adapts this idea to unconditional graph generation: we remove every training graph containing the target and measure its emission, not rule acquisition; to our knowledge, prior evaluations of serialized-graph generators do not use this fixed-target removal controls.

n-gram accounts and generative families. Transformer next-token predictions are often approximated by training-data n-gram rules (Nguyen, 2024), and unbounded n-gram models predict next tokens well (Liu et al., 2024); this motivates our finite-context n-gram baselines over DFS edge tokens, including Witten–Bell smoothing (Witten & Bell, 1991; Chen & Goodman, 1999). Absorbing-state discrete diffusion (Austin et al., 2021) (on graphs, GraphARM; Kong et al., 2023) is exactly linked to any-order autoregressive modeling (Hoogeboom et al., 2022; Ou et al., 2025), one reason different generative families can share support statistics; our DiGress comparison (Appendix G) does not test this link, since DiGress corrupts rather than masks node and edge categories.

3 Setup: Support, Serializations, and Measurement

3.1 Support under a Serialized-Graph Generator

Graphs, strings, and invalid outputs. Let $\mathcal{G}$ be the set of isomorphism classes of finite labeled graphs, $\mathcal{X}$ the set of finite token strings, and $\bar{\mathcal{G}} = \mathcal{G} \cup \{\perp\}$, with $\perp$ an invalid output. A deterministic serializer ϕ maps each stored graph to one string; a strict parser $\psi : \mathcal{X} \to \bar{\mathcal{G}}$ returns the decoded graph's class, or $\perp$ for a malformed string, without repair. We idealize the data as a distribution P on whole graphs ($P(\perp) = 0$), with $\psi_{\#}P_X = P$ for the stored-string distribution P_X. A *generator*—a trained model together with its decoding rule—defines a distribution Q_X over strings. Decoding induces the graph distribution Q; since different strings can decode to the same graph, $Q(G)$ sums over all strings that decode to G:

$$Q = \psi_{\#}Q_X, \qquad Q(G) = \textstyle\sum_{x:\,\psi(x)=G} Q_X(x) \quad (G \in \bar{\mathcal{G}}),$$

whereas $Q_X(\phi(G))$ is the probability of generating one particular encoding of G.

Definition 1 (Support)**.** For a pattern H and distribution R on $\bar{\mathcal{G}}$, $s_H(R) = R(\{G \in \mathcal{G} : H \sqsubseteq G\})$, with $\sqsubseteq$ a label-preserving, possibly non-induced subgraph monomorphism; $\perp$ contains no pattern.

Lemma 2 (Support is an event probability)**.** *For distributions P, Q on $\bar{\mathcal{G}}$ and any pattern family $\mathcal{H}$,* $\sup_{H\in\mathcal{H}} |s_H(Q) - s_H(P)| \le \mathrm{TV}(P, Q) = \sup_{A\subseteq\bar{\mathcal{G}}} |P(A) - Q(A)|$.

Lemma 3 (Validity split)**.** *Let $q = Q(\perp) < 1$ and $Q_{\mathrm{v}} = Q(\cdot \mid \mathcal{G})$. For every pattern H, $s_H(Q) = (1-q)\, s_H(Q_{\mathrm{v}})$ and $s_H(Q) - s_H(P) = (1-q)\big[s_H(Q_{\mathrm{v}}) - s_H(P)\big] - q\, s_H(P)$.*

We call $s_H(Q)$ *total-map* and $s_H(Q_{\mathrm{v}})$ *valid-only* support, and $s_H(Q_{\mathrm{v}}) - s_H(P)$ *in-support error*.

Corollary 4 (Support agreement does not identify the generator)**.** *Let $\hat{P}_{\mathrm{tr}}$ be the empirical distribution of the training graphs and R any distribution on graphs outside the training set. For $\lambda \in [0, 1]$, $M_\lambda = (1-\lambda)\hat{P}_{\mathrm{tr}} + \lambda R$ places mass λ outside the training set, yet* $\sup_H |s_H(M_\lambda) - s_H(\hat{P}_{\mathrm{tr}})| \le \lambda$.

These standard facts (Cover & Thomas, 2006) fix estimands and are not contributions; Appendix A proves them and adds Lemmas 6 (parsing does not increase divergence) and 7 (rank preservation). Consequences: (a) invalid outputs multiply every valid-only support by $1 - q$ (Lemma 3), leaving ranks unchanged; we report invalid mass separately from in-support error. (b) Agreement is expected of any generator TV-close to P (Lemma 2) and can also arise from replay, which need not be TV-close. For a uniform training sample of our size and any 253 patterns fixed independently of it, all empirical supports lie within 0.0012 of those under P with probability at least 0.95. This illustrates replay's behavior; the bound does not directly apply to our data-selected panel (Remark 8). By Corollary 4, support agreement alone cannot identify how outputs are produced. (c) Likelihood-based (Pinsker) bounds are too loose to explain the observed agreement (Appendix K.3).

3.2 Data, Serializations, Models, and Sampling

Data. We use all 3,746,620 molecules of PCQM4Mv2 (Hu et al., 2021) as heavy-atom graphs labeled by element and Kekulé bond order; hydrogens (implicit), charges, stereochemistry, and

aromaticity are not represented. A seed-42 random 90/5/5 split, not the official OGB-LSC split, gives 3,371,958 training graphs and 187,331 each for validation and test.

DFS-code serialization. Following gSpan (Yan & Han, 2002), a depth-first traversal emits one token per edge event $\langle i, j, \ell_i, \ell_j, \ell_{ij} \rangle$ (discovery indices; atom and bond labels). Ring-closing edges appear twice, once per endpoint, without a direction flag. The string is deterministic but, unlike the minimum DFS code of gSpan and GraphGen (Goyal et al., 2020), not isomorphism-canonical.

Construction-sequence serialization (DGMG-style). Target removal uses node/edge-addition sequences (Li et al., 2018). Atoms are added grouped by element symbol, ties broken by stored atom index, each followed by its bonds to earlier atoms, so only 16.0% of training sequences keep the partial graph connected at every step. The sequence is likewise deterministic and not isomorphism-canonical. Support (Section 4) and removal (Section 5) use different serializations and separately trained models; we compare no numbers across them.

Models and sampling. All neural models in Sections 4–5 are LLaMA-architecture decoders (Touvron et al., 2023) with 12 layers, width 768 and 12 heads, trained from scratch by next-token prediction. The *headline DFS-code model* (132.2M parameters) is one training run (peak learning rate 10^{-5}, 50 epochs). Each target-removal model (113.4M parameters; same body, smaller vocabulary) is trained for 2.27B tokens (Appendix C). Sampling is unconditional. For support, the primary decoder is *pure ancestral sampling* (temperature 1, no truncation), with 4,096 attempts for each of five *generation seeds* (fixed weights, not training replicates). Target-removal sampling uses temperature 1, top-k 50, and top-p 0.95 (Fan et al., 2018; Holtzman et al., 2020).

3.3 Measurement

Validity and novelty. An output is *strict-valid* if it parses fully, without repair, into a connected labeled graph (structural, not chemical; Appendix D); invalid outputs map to $\perp$ and stay in the raw denominator. A strict-valid DFS output is *exact-string novel* if its string is not a training string; as it can still be a training graph, target removal uses *isomorphism novelty*: no labeled isomorphism to any graph of the original, unfiltered training split.

Fixed panel and exact counting. One panel of 253 labeled patterns with 2–8 atoms (248 acyclic; 5 contain a benzene ring) was mined once with gSpan (gBolt; Zhou, 2019) at minimum support 0.1 on 10,000 training molecules. Reference supports $s_H(P)$, computed on 200,000 strict-valid training graphs, span 0.095–0.994: this panel covers only frequent patterns. For empirical comparisons, P denotes the stated finite reference distribution, rather than an unknown population distribution. A second panel, mined from the same molecules at minimum support 0.01, has 3,754 patterns (142 cyclic), includes the 253 and extends to rarer ones (reference supports 0.0078–0.994); only Figure 1, two columns of Table 1 and Appendices E.3 and F use it. Every generated graph is tested against every pattern (VF2 subgraph monomorphism; Cordella et al., 2004; Hagberg et al., 2008); generated sets are never re-mined, since re-mining censors near-threshold patterns differently per generator (Table 8 in Appendix E.1). From the replica, isomorphism, DiGress, and TUDataset checks, which re-mine (Appendices E–H), we quote only Spearman's ρ, which censoring barely moves.

Conventions and metrics. The *total map* divides by all raw attempts; *valid-only*, our primary support convention, divides by strict-valid outputs; *novel-only* keeps exact-string-novel ones. With $\Delta s_H = \hat{s}_H - s_H(P)$ ($\hat{s}_H$: generated support under the convention used), we report the panel mean $\overline{|\Delta s|}$, its $s_H(P)$-weighted version, Kendall's τ_a, and Spearman's ρ.

Reference controls. *Real-data resamples* (training graphs, or test graphs matching no training string) provide same-n empirical reference scores; the test reference also reflects the shift introduced by its selection. A specified *bridge-only fragment-recombination* baseline cuts two training molecules at bridge bonds of equal order and rejoins exchanged fragments. We also use *finite-context n-gram models over DFS edge tokens* ($k = 3, 5$; exact counts); Section 6 adds interpolated Witten–Bell models. Baseline outputs are rejection-sampled to be strict-valid, exact-string novel, and unique, so we compare them only with the Transformer's valid-only and novel-only outputs.

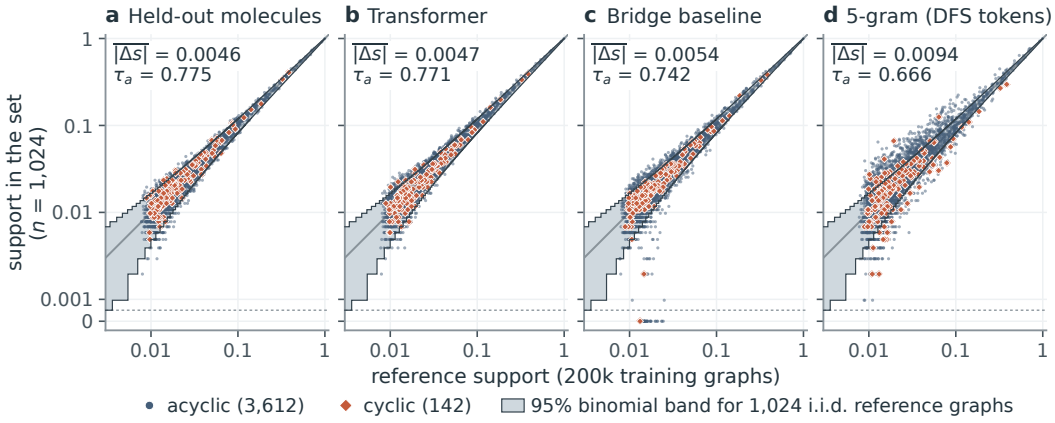


Figure 1: Generated versus reference support of the 3,754 patterns of the minimum-support-0.01 panel (PCQM4Mv2, DFS codes). Each panel: the $n = 1{,}024$ set closest to its Table 1 means (last two columns; annotated). Shaded band: pointwise (not simultaneous) 95% binomial sampling interval for 1,024 i.i.d. graphs, reference supports held fixed; diamonds: 142 cyclic patterns; row "0": zero support. (a) Held-out molecules; (b) Transformer (pure ancestral sampling); (c) bridge-only baseline; (d) exact-count 5-gram (DFS edge tokens). All sets exclude exact training strings.

Table 1: Support reproduction at $n = 1{,}024$ graphs per set (PCQM4Mv2, DFS codes; 200,000-graph training reference). Columns 3–7: the 253-pattern panel; last two: the 3,754-pattern panel mined from the same molecules at minimum support 0.01 (same sets and procedure). Resamples: strict-valid training or string-novel held-out test graphs. Transformer: uniform subsamples of one trained model's pure-sampling outputs per generation seed. Baselines: rejection-sampled (Section 3.3). Five resampling or generation seeds per row; ±: SD over single sets; signed: mean Δs.

		253 patterns (min. support 0.1)					3,754 (0.01)	
Source	Kept outputs	$\overline{\lvert\Delta s\rvert}\downarrow$	mass-wtd. ↓	$\tau_a\uparrow$	$\rho\uparrow$	signed	$\overline{\lvert\Delta s\rvert}\downarrow$	$\tau_a\uparrow$
Resample: training	all	0.0083 ± 0.0015	0.0087	0.910 ± 0.010	0.985	+0.001	0.0039	0.791
Resample: held-out	string-novel	0.0121 ± 0.0027	0.0136	0.883 ± 0.009	0.978	+0.005	0.0046	0.775
Transformer	strict-valid	0.0101 ± 0.0018	0.0108	0.893 ± 0.016	0.980	+0.001	0.0043	0.778
Transformer	string-novel	0.0112 ± 0.0016	0.0119	0.881 ± 0.016	0.975	+0.003	0.0047	0.771
Bridge baseline	string-novel	0.0138 ± 0.0015	0.0142	0.856 ± 0.006	0.965	+0.002	0.0054	0.742
5-gram	string-novel	0.0255 ± 0.0031	0.0270	0.765 ± 0.021	0.912	+0.000	0.0094	0.666
3-gram	string-novel	0.0454 ± 0.0024	0.0481	0.643 ± 0.006	0.813	−0.012	0.0146	0.545

4 Observation: Generated Graphs Reproduce Substructure Support

We first ask how closely an autoregressive model trained only on whole-graph strings, without explicit substructure supervision, reproduces substructure support: the fraction of graphs containing a given pattern. We study PCQM4Mv2 graphs serialized as DFS codes and compute supports by exact subgraph matching on our primary panel of 253 patterns (Section 3.3); we report the mean absolute difference $\overline{\lvert\Delta s\rvert}$ between generated and reference supports and their Spearman rank correlation ρ across patterns. As finite real samples also deviate from the reference supports, we compare generated samples with same-size real-data resamples. Unless stated otherwise, each evaluation sample contains $n = 1{,}024$ graphs, and Transformer results use a single trained model under pure ancestral sampling. Generation seeds are sampling streams of this fixed model, not independent training runs.

4.1 Frequent-Pattern Support Is Close to That of Same-Size Real Samples

Strict-valid Transformer outputs achieve $\overline{|\Delta s|} = 0.0101$ ($\rho = 0.980$; Table 1), close to the training-resample error (0.0083) and the expected error of drawing n graphs i.i.d. from the reference distribution (0.0095; Table 6 in Appendix D.3). About a quarter of strict-valid outputs are exact training strings. Excluding them gives an error of 0.0112 ($\rho = 0.975$), comparable to a held-out resample that likewise excludes training strings (0.0121). Close agreement thus survives removing exact-string copies. On the larger 3,754-pattern panel, which extends to rarer patterns, the error stays close to the real-data resamples and the i.i.d. benchmark, including within each support stratum (Figure 1; Table 1; Appendix E.3). For this model, corpus, and decoder, strict-valid outputs reproduce the evaluated supports about as closely as same-size real samples.

Three additional checks examine variation across training runs, graph-level copying, and the resolution of support rankings. These checks use a separate re-mined protocol (temperature 1, top-k 50, top-p 0.95), for which we report only Spearman's ρ (Section 3.3); its values are not directly comparable with Table 1. Four independently trained replicas achieve correlations similar to the main model's under this protocol (0.945–0.952 versus 0.950 at $n = 700$; Table 11 in Appendix E.2). About 5% of string-novel outputs still represent a training graph through a different DFS string; excluding them changes ρ against held-out references only from 0.965 to 0.966 at matched $n = 2{,}597$. High rank agreement therefore persists after graph-level copies are removed, although both correlations remain below that of a same-size held-out resample (0.985; Table 10 in Appendix E.2).

Finally, in the re-mined panels, the lowest-support tenth of reference patterns occupies a band about as wide as the sampling standard error; rankings within it are unstable even for real-data resamples, so this evaluation says little about their relative order (Table 9 in Appendix E.1).

4.2 Invalid Outputs and Decoding

The preceding results condition on strict validity; we now examine how invalid attempts and the decoder affect support agreement. In the full generation pools (4,096 attempts per seed), a fraction $\hat{q} = 0.142$ of pure-sampling attempts fails the strict parser. The total map keeps these attempts in the denominator as containing no pattern, so it scales every valid-only support by $1 - q$, preserving ranks but changing magnitudes (Lemma 3). The mass-weighted mean absolute error increases from 0.0067 (valid-only) to 0.0434 (total-map). Padding real-data samples with empty graphs to this invalid fraction yields errors of 0.0429–0.0453 (Appendix D.3). Invalidity can thus account for most of the total-map error, so we report invalid mass separately from support error among valid outputs.

Decoding also affects valid-output supports. At temperature 1, nucleus truncation with top-p 0.95, alone or with top-k 50, over-represents about 96% of the 253 patterns, versus about half under pure or top-k-only sampling. Across these four decoders, valid-only ρ remains at 0.98–0.99, while the unweighted $\overline{|\Delta s|}$ ranges from 0.0057 to 0.0256 (Table 7 in Appendix E). These full-pool comparisons use decoder-dependent numbers of valid outputs. Still, the broad over-representation under top-p shows that high rank agreement can coexist with systematic magnitude errors.

4.3 A Bridge-Only Baseline Reaches Comparable Agreement

Closeness to the data distribution implies support agreement (Lemma 2), but support agreement can also arise from replaying training graphs and does not identify the generator (Corollary 4). We therefore ask whether simple non-neural generators reproduce them too. The specified bridge-only baseline (Appendix F) reaches $\overline{|\Delta s|} = 0.0138$, comparable to the Transformer's string-novel outputs (0.0112) and held-out resamples (0.0121; Table 1). The exact-count n-gram models achieve substantial rank agreement ($\rho = 0.912$ for $k = 5$), but their errors are about 2.3 ($k = 5$) and 4 ($k = 3$) times that of the Transformer's string-novel outputs. They also underestimate support on average over the five ring-containing patterns. We do not read the small gap to the bridge baseline as Transformer superiority: the comparison is post hoc, uses one trained Transformer, and only bridge outputs must pass an RDKit filter (RDKit, 2026). Aggregate support agreement, even when comparable to real-data resamples, therefore does not identify how graphs are generated.

Two ancillary checks further illustrate the limits of rank agreement: it is high even in runs generating only training strings on five small TUDataset benchmarks (Morris et al., 2020; Appendix H) and,

for a DiGress diffusion model (Vignac et al., 2023), within the range of same-size reference samples (PCQM4Mv2 subset, $n = 138$; Appendix G), so it is not exclusive to autoregressive models.

5 TEST: GENERATING TARGETS WITHHELD FROM TRAINING

Since replay (Corollary 4) and the bridge-only baseline (Section 4.3) can also yield support agreement, we ask whether a model generates a specified target absent from its training graphs. We remove every training graph containing the target, retrain from scratch, and count unconditional samples containing it. Target removal and hit detection are defined on graphs; here, we apply this test to construction-sequence models trained separately from the DFS-code model of Section 4.

5.1 DESIGN

Targets. C_3 is a triangle of carbon atoms joined by single bonds. Without C_3-containing training graphs, neither copying retained graphs nor bridge-only recombination of their fragments can produce it (Section 5.3). P_6 is a path of six carbon atoms joined by five single bonds, each a *bridge* of the host graph (an edge on no cycle). It can be built by such recombination and is an acyclic contrast of similar frequency in the original training data. Containment follows Definition 1, plus the bridge condition for P_6; we count graphs containing the target, not occurrences within a graph. Both targets were chosen by fixed frequency rules (Appendix I.1): C_3 before any model was trained, and P_6 after the C_3 models started training but before any C_3 or P_6 outcome was inspected.

Removal and controls. The *target-withheld* condition removes every training graph containing the target: 193,874 of 3,371,958 graphs for C_3 and 178,620 for P_6. Other labeled triangles remain in the C_3-withheld set, and six-carbon single-bond paths with a non-bridge edge remain in the P_6-withheld set (Appendix I.2). To control for the amount of removed data, the *matched-removal* condition removes the same number of target-free graphs, matched on sequence length, while retaining every target-containing graph. This controls training-set size and closely matches sequence lengths, but not all changes in graph composition. *Full-data* models remove nothing.

Models and sampling. Models are 113.4M-parameter Transformers on construction sequences. Each condition has three training seeds; same-seed models share initialization, and the three full-data models serve both targets (15 models in total). Each model draws 163,840 unconditional, unfiltered samples with the decoder of Section 3.2. Sampling streams use common random numbers across models and are not independent training replicates; we report each training seed separately.

Endpoint and suppression. A *hit* is a strict-valid output whose graph contains the target; it is *novel* if not labeled-isomorphic to any graph of the original, unfiltered training split. All models share one budget of *raw attempts*, invalid ones included, so hit-count ratios equal hit-rate ratios. Per training seed, we compare target-withheld with matched-removal hits (inverse ratio: fold suppression). The C_3 generation endpoint was fixed after training began, before any C_3 outcome was inspected. The earlier teacher-forced primary endpoint was not formally evaluated (Appendix I.5).

5.2 RESULTS

We first discuss C_3. Models whose training data contain no C_3 nevertheless produce C_3-containing outputs across all three training seeds, with 15, 22, and 28 hits, respectively. Relative to matched removal, these counts correspond to 430- to 800-fold suppression without eliminating observed emission (Figure 2a; Table 17 in Appendix I.7). All 65 hits are distinct labeled graphs, and none is labeled-isomorphic to any original training graph, retained or removed. The low hit rates do not reflect a collapse in structural validity: across all 15 generation sets, 97.3–97.7% of attempts are strict-valid (structural, not chemical).

We next discuss P_6. Suppression is much weaker than for C_3: target-withheld models produce 1,057, 1,100, and 1,037 hits (938, 964, and 918 novel), corresponding to about tenfold suppression relative to matched removal (Figure 2b). Of these hits, 119, 136, and 119, respectively, are labeled-isomorphic to withheld training graphs. Across the three models, these hits represent 299 distinct labeled graphs, 60 of which are reproduced by more than one model. Every P_6-containing training graph was removed, so a non-novel hit must match a withheld graph (Appendix I.4). These hits thus regenerate removed graphs; they do not replay retained ones.

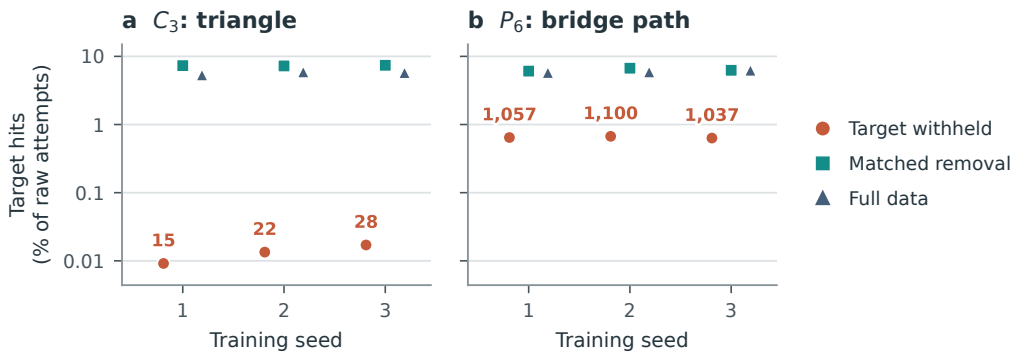


Figure 2: Emission of withheld targets (PCQM4Mv2; 113.4M-parameter construction-sequence Transformers; decoder in Section 3.2). Each marker (one trained model) shows the percentage of its 163,840 raw attempts that are strict-valid and contain the target, novel or not (log scale). Labels give target-withheld hit counts. Panels differ in target size, bridge condition, and removed graphs.

5.3 What the Result Rules Out, and What It Does Not

Replay and bridge-only recombination. We assess two candidate explanations illustrated in Section 4 against the observed C_3 emissions. No retained training graph contains C_3, excluding replay of the retraining data, and no C_3 hit is labeled-isomorphic to any original training graph. The bridge-only baseline of Section 4.3 belongs to a class that cannot produce C_3 after removal:

Lemma 5 (Bridge-only recombination cannot create C_3)**.** *Let $\mathcal{D}$ be a set of training graphs none of which contains C_3. Suppose every output G of a generator is obtained from a disjoint union of fragments $F_1, \ldots, F_m$, each a labeled subgraph of some graph in $\mathcal{D}$, by adding labeled edges each of which is a bridge of G. Then no output contains C_3.*

Proof idea (Appendix I.9): no cycle contains a bridge, so every triangle of G lies inside one fragment, and subgraphs of C_3-free graphs are C_3-free. The baseline belongs to this class because its rejoining bond is a bridge. In contrast, P_6 can be produced by this class from retained training fragments (Appendix I.9). Thus, emission of either withheld target excludes replay of retained graphs, whereas C_3 emission additionally excludes the specified bridge-only class.

Truncated decoding. Under untruncated softmax sampling with finite logits, every sequence within the vocabulary and length limit has positive probability, so positive probability alone is uninformative. Here, all Transformer conditions use the same per-step truncation (top-k 50, then top-p 0.95); reported hits are actual emissions under it within a fixed sampling budget. This does not imply that their complete sequences are probable: token-level truncation still permits rare sequences.

Local token statistics. The lemma excludes a specified graph-recombination class, not local token mechanisms. In a control designed after the Transformer outcomes and fixed before it ran (Appendix I.8), an exact-count 5-gram over construction-sequence tokens, trained on the C_3-withheld set and sampled with the same decoder and raw-attempt budget, emits 58 C_3 hits, 55 novel. Every 5-token window in these hits occurs in the retained training data. This demonstrates that local token statistics alone can produce C_3-containing outputs in this control.

The 5-gram retains 44% of its matched-removal count (3-gram 116%, 8-gram 21%). Because these models have much lower structural validity, we compare removal effects relative to each generator's own matched-removal control. In these comparisons, Transformer emission is much more strongly suppressed, to 0.13–0.23% of matched-removal counts. The outputs also differ in RDKit sanitization: 55 of the 65 Transformer hits pass, compared with 12 of the 58 5-gram hits (RDKit, 2026). These comparisons are descriptive and limited to the evaluated exact-count k-grams.

Local edits and general rules. Ten of the 65 C_3 hits are one local change tested in Appendix J away from a retained training graph (Figure 3). The other 55 have no single-change witness, which does not exclude multiple or other local changes. Nor do the results establish a general ring-forming rule or that cyclicity causes the C_3/P_6 rate gap: the targets differ in size, bridge condition, and removed graphs. For the construction-sequence models and decoder evaluated here, removal strongly

suppresses C_3 emission without eliminating it; replay and the specified bridge-only recombination class cannot explain the C_3 hits, whereas local token mechanisms and local edits remain possible.

6 Supporting Analyses: Likelihood and Other Generators

To assess whether held-out likelihood can replace direct support evaluation (Section 4), we compare support errors of models with similar negative log-likelihood (NLL, nats per DFS token, 20,000 validation molecules) under pure ancestral sampling (Appendix K, Figure 4).

Within one training recipe. We test whether lower NLL is associated with lower support error as model size and training length vary. In 48 checkpoints of eight Transformer runs (10.5M–451M parameters), NLL varies with both, so we correlate NLL with each error measure after adjusting for model size and training length (Spearman; 95% bootstrap intervals over runs). Mass-weighted total-map error (here Pearson 0.9999 with invalid mass q) has no detectable residual association (-0.32, $[-0.80, 0.48]$); valid-only error, which isolates in-support error (Lemma 3), does (0.92, $[0.74, 0.98]$): within this recipe, lower NLL goes with lower in-support error beyond training length and size. This is an association, not a causal effect, and NLL is confounded with mean generated graph size here (Spearman -0.92).

Across model classes. If NLL fixed support error, matched-NLL generators would share it. At nearby NLL, a Witten–Bell model on five preceding DFS edge tokens (not the exact-count 5-gram of Section 4.3) is comparable to Transformer checkpoints (valid-only error 0.016 at NLL 1.449; Table 23 in Appendix K.2), but two LSTMs (Hochreiter & Schmidhuber, 1997) show higher support error (exploratory; Appendix K.2). We compare NLL and mean generated node count as predictors of this higher support error. Predicting support error for 16 LSTM checkpoints from their four nearest Transformer ones gives a mean $|\log\text{-ratio}|$ (observed/predicted error) of 1.19 by NLL but 0.32 by mean node count: the excess follows generated size, as LSTM graphs are smaller. Similar NLL values do not necessarily imply similar support error. Therefore, NLL cannot replace direct support evaluation. Single training seeds and differing training recipes also preclude an architecture ranking.

7 Discussion and Conclusion

On PCQM4Mv2, both statistical reproduction of substructure support and generation of withheld targets appear. Under pure ancestral sampling, DFS-code outputs reproduce the supports of 253 frequent substructures about as closely as same-size real-data resamples, also without exact training strings. Under top-k/top-p sampling, C_3-withheld construction-sequence models still emit strict-valid C_3-containing outputs, none labeled-isomorphic to an original training graph, but removal suppresses C_3 430- to 800-fold and P_6 about tenfold relative to matched removal, far more than in exact-count k-grams trained on the same sets.

What the evidence does not show. It does not identify a mechanism: a bridge-only recombination baseline reaches comparable agreement (Section 4.3), and the C_3 result excludes that class but not local token statistics, which suffice for exact-count k-grams to emit C_3, or local edits (Section 5.3). Nor does it show acquisition of general compositional rules, a cyclicity effect behind the C_3/P_6 rate gap, or serialization invariance. Held-out likelihood tracks in-support error only within one training recipe (Section 6), and transfer beyond PCQM4Mv2 is open.

Limitations. The main tests use one corpus (random, non-official split), one target of each kind, three training seeds per removal condition and small C_3 counts. Each headline result uses one decoder; strict validity is structural, not chemical. The C_3 generation endpoint was fixed after training began, before any C_3 outcome was inspected (Appendix I.5). The headline DFS-code model is one training run; four further runs served only to check rank agreement. Next steps: other generator classes trained on the removal sets, more targets, and removal under both serializations.

On PCQM4Mv2, whole-graph sequence training thus reproduces substructure-level statistics that simpler generators also reach, while target removal shows more: the models generate specified structures absent from training, the C_3 outputs beyond bridge-only recombination, yet suppress them far more strongly than local token statistics do. Neither result identifies a mechanism; the representations and rules behind this substructure generalization remain open.

AI USE STATEMENT

The research questions, conceptual framework, and core methodology were conceived by the authors, who also wrote the core codebase, including graph serialization, model training, and sampling. Generative AI assistants were used to provide feedback on the experimental design; to implement and run, under the authors' direction, additional analyses such as the finite-context k-gram control of Section 5.3; to extend author-written code and to implement and review evaluation and provenance-validation scripts; to organize experimental evidence and prepare result figures and tables; and to draft and revise manuscript text from the authors' verified results. They were also used as a secondary check of standard probabilistic identities and algebraic derivations in the evaluation framework, which the authors verified independently. All AI-assisted code, numerical outputs, experimental suggestions, mathematical feedback, and text were critically examined and verified by the authors against recorded experimental artifacts and, where applicable, independent derivations. The authors take full responsibility for the entirety of this work, including all final experimental decisions, mathematical claims, code, results, and text.

REPRODUCIBILITY STATEMENT

Data and models. We use all 3,746,620 PCQM4Mv2 molecules (Hu et al., 2021) with a seed-42 random 90/5/5 split, not the official OGB-LSC split. Section 3.2 and Appendices B and C specify the data, both serializations, the models and the sampling of the headline support measurements (Section 4). The headline DFS-code model is one training run (training seed 42). Four replicas repeat its recipe; the one with training seed 42 shares the headline run's seed and data order but ends with different weights. Tables 4 and 5 list model configurations and training recipes, including token budgets and training runs. The headline and target-removal runs record PyTorch 2.7.1 (CUDA 12.6 build) and Transformers 4.57.3; DFS-code training uses TRL 0.21.0.

Sampling and measurement. Appendix K specifies the sampling of Section 6, and Appendix I.4 that of Section 5; the other analyses state their decoder, n and seeds where they appear. Section 3.3 and Appendix D define the strict parsers, both novelty tests, the two frozen pattern panels and the training reference.

Target removal and compute. Appendix I.2 documents how the removal sets were built and verified. Every training and sampling job for the autoregressive Transformers used one NVIDIA A100 80GB GPU. Training took about 176 GPU-hours for the eight DFS-code runs (headline run, four replicas and three other sizes) and 190.7 GPU-hours for the 15 models of the target-removal experiments. Sampling one target-removal model took 32–34 minutes (measured on the six P_6-specific models). These totals exclude the LSTM and learning-rate-probe runs of Appendix K. Code, configurations, the trained checkpoints (all but the autoregressive comparator of Appendix G), the two frozen pattern panels, the removal index files and generated samples will be released in an anonymized supplement. The released pipeline regenerates every table and figure on CPU from the released samples and run records.

ETHICS STATEMENT

This work uses only public graph datasets, PCQM4Mv2 (Hu et al., 2021) and TUDataset benchmarks (Morris et al., 2020), and involves no human subjects or personal data. Our validity criterion is structural, not chemical (Section 3.3), and we make no claim that generated graphs correspond to stable, synthesizable or useful molecules. The models generate graphs unconditionally, without property conditioning or optimization toward any activity, and the two withheld targets are small generic carbon substructures. We therefore judge the added dual-use risk to be limited. The use of AI assistants is disclosed in the AI Use Statement.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information*

Processing Systems, volume 34, pp. 17981–17993, 2021.

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

Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alexander J. Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21 (suppl_1):i47–i56, 2005. doi: 10.1093/bioinformatics/bti1007.

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

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650. USENIX Association, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. Quantifying memorization across neural language models. In *International Conference on Learning Representations*, 2023.

Dexiong Chen, Markus Krimmel, and Karsten Borgwardt. Flatten graphs as sequences: Transformers are scalable graph generators. In *Advances in Neural Information Processing Systems*, volume 38, pp. 69071–69109, 2025. doi: 10.52202/085713-2323.

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393, 1999. doi: 10.1006/csla.1999.0128.

Luigi P. Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1367–1372, 2004. doi: 10.1109/TPAMI.2004.75.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, Hoboken, NJ, second edition, 2006. ISBN 978-0-471-24195-9.

Asim Kumar Debnath, Rosa L. Lopez de Compadre, Gargi Debnath, Alan J. Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry*, 34(2):786–797, 1991. doi: 10.1021/jm00106a046.

Paul D. Dobson and Andrew J. Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology*, 330(4):771–783, 2003. doi: 10.1016/S0022-2836(03)00628-4.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1082.

Daniel Flam-Shepherd, Kevin Zhu, and Alán Aspuru-Guzik. Language models can learn complex molecular distributions. *Nature Communications*, 13:3293, 2022. doi: 10.1038/s41467-022-30839-x.

Nikhil Goyal, Harsh Vardhan Jain, and Sayan Ranu. GraphGen: A scalable approach to domain-agnostic labeled graph generation. In *Proceedings of The Web Conference 2020*, pp. 1253–1263. ACM, 2020. doi: 10.1145/3366423.3380201.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy 2008)*, pp. 11–15, 2008. doi: 10.25080/TCWV9851.

Christoph Helma, Ross D. King, Stefan Kramer, and Ashwin Srinivasan. The predictive toxicology challenge 2000–2001. *Bioinformatics*, 17(1):107–108, 2001. doi: 10.1093/bioinformatics/17.1.107.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.

Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. In *International Conference on Learning Representations*, 2022.

Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. OGB-LSC: A large-scale challenge for machine learning on graphs. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020. doi: 10.1613/jair.1.11674.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2323–2332. PMLR, 2018.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4839–4848. PMLR, 2020.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10362–10383. PMLR, 2022.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*, 2020.

Najoung Kim and Tal Linzen. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9087–9105. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.731.

Lingkai Kong, Jiaming Cui, Haotian Sun, Yuchen Zhuang, B. Aditya Prakash, and Chao Zhang. Autoregressive diffusion model for graph generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17391–17408. PMLR, 2023.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2873–2882. PMLR, 2018.

Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter W. Battaglia. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, William L. Hamilton, David Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. In *First Conference on Language Modeling (COLM)*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. TUDataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+)*, 2020.

Timothy Nguyen. Understanding transformers via n-gram statistics. In *Advances in Neural Information Processing Systems*, volume 37, pp. 98049–98082, 2024. doi: 10.52202/079017-3112.

Leslie O'Bray, Max Horn, Bastian Rieck, and Karsten Borgwardt. Evaluation metrics for graph generative models: Problems, pitfalls, and practical solutions. In *International Conference on Learning Representations*, 2022.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *International Conference on Learning Representations*, 2025.

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

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC20)*, pp. 1–16. IEEE, 2020. doi: 10.1109/SC41405.2020.00024.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. DeepSpeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 3505–3506. ACM, 2020. doi: 10.1145/3394486.3406703.

RDKit. RDKit: Open-source cheminformatics. `https://www.rdkit.org`, 2026. Release 2026.03.1; concept DOI 10.5281/zenodo.591637.

Philipp Renz, Dries Van Rompaey, Jörg Kurt Wegner, Sepp Hochreiter, and Günter Klambauer. On failure modes in molecule generation and optimization. *Drug Discovery Today: Technologies*, 32–33:55–63, 2019. doi: 10.1016/j.ddtec.2020.09.003.

Ida Schomburg, Antje Chang, Christian Ebeling, Marion Gremse, Christian Heldt, Gregor Huhn, and Dietmar Schomburg. BRENDA, the enzyme database: Updates and major new developments. *Nucleic Acids Research*, 32(suppl_1):D431–D433, 2004. doi: 10.1093/nar/gkh081.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12(77): 2539–2561, 2011. URL `http://jmlr.org/papers/v12/shervashidze11a.html`.

Rylee Thompson, Boris Knyazev, Elahe Ghalebi, Jungtaek Kim, and Graham W. Taylor. On evaluation metrics for graph generative models. In *International Conference on Learning Representations*, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer, New York, 2009. doi: 10.1007/b13794.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete denoising diffusion for graph generation. In *International Conference on Learning Representations*, 2023.

Nikil Wale, Ian A. Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14(3):347–375, 2008. doi: 10.1007/s10115-007-0103-5.

David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988. doi: 10.1021/ci00057a005.

Ian H. Witten and Timothy C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4): 1085–1094, 1991. doi: 10.1109/18.87000.

Masatsugu Yamada and Mahito Sugiyama. Molecular graph generation by decomposition and reassembling. *ACS Omega*, 8(22):19575–19586, Jun 2023. doi: 10.1021/acsomega.3c01078. URL `https://doi.org/10.1021/acsomega.3c01078`.

Xifeng Yan and Jiawei Han. gSpan: Graph-based substructure pattern mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM)*, pp. 721–724. IEEE, 2002. doi: 10.1109/ICDM.2002.1184038.

Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. GraphRNN: Generating realistic graphs with deep auto-regressive models. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5708–5717. PMLR, 2018.

Keren Zhou. gBolt: A fast, memory efficient, and light-weight implementation for the gSpan algorithm in data mining. GitHub repository, `https://github.com/Jokeren/gBolt`, 2019.

A PROOFS

We prove the statements of Section 3.1 to fix constants and conventions, and state two further standard facts in the same notation. Lemma 6 carries string-level divergences over to graphs, which the likelihood bound of Appendix K.3 needs. Lemma 7 shows that support errors below half the gap between two reference supports cannot swap those two patterns; high rank agreement does not imply small errors (Section 4.2), so we report support magnitudes as well as rank statistics.

Lemma 6 (Parsing does not increase divergence)**.** *For string distributions P_X, Q_X,* $\mathrm{TV}(\psi_{\#}P_X, \psi_{\#}Q_X) \le \mathrm{TV}(P_X, Q_X)$ *and* $\mathrm{KL}(\psi_{\#}P_X \,\|\, \psi_{\#}Q_X) \le \mathrm{KL}(P_X \,\|\, Q_X)$.

Lemma 7 (Rank preservation)**.** *Let $\mathcal{H}$ be finite with $|\mathcal{H}| \ge 2$, and let $\varepsilon = \sup_{H\in\mathcal{H}} |s_H(Q) - s_H(P)|$. If $s_H(P) - s_{H'}(P) > 2\varepsilon$, then $s_H(Q) > s_{H'}(Q)$. Hence Kendall's τ_a between $(s_H(P))_{H\in\mathcal{H}}$ and $(s_H(Q))_{H\in\mathcal{H}}$ satisfies $\tau_a \ge 1 - 2\pi^{+}_{2\varepsilon} - t_P$, where t_P is the fraction of pattern pairs tied under P and $\pi^{+}_{2\varepsilon}$ the fraction with $0 < |s_H(P) - s_{H'}(P)| \le 2\varepsilon$.*

The graph space $\mathcal{G}$ (isomorphism classes of finite graphs with labels from finite alphabets) and the string space $\mathcal{X}$ (finite strings over a finite vocabulary) are countable, so all distributions are discrete and every subset is an event. We use $\mathrm{TV}(P, Q) = \sup_A |P(A) - Q(A)| = \frac{1}{2}\sum_G |P(G) - Q(G)|$ and write $A_H = \{G \in \mathcal{G} : H \sqsubseteq G\}$, so that $s_H(R) = R(A_H)$ for every distribution R on $\bar{\mathcal{G}}$; $\perp$ lies outside every A_H.

Proof of Lemma 2. For each pattern H, $|s_H(Q) - s_H(P)| = |Q(A_H) - P(A_H)| \le \sup_{A \subseteq \bar{\mathcal{G}}} |Q(A) - P(A)| = \mathrm{TV}(P, Q)$; take the supremum over $H \in \mathcal{H}$. On a countable space the supremum over events is attained at $A = \{G : P(G) > Q(G)\}$ and equals $\frac{1}{2}\sum_{G \in \bar{\mathcal{G}}} |P(G) - Q(G)|$. The constant cannot be improved: moving mass t from $\bar{\mathcal{G}} \setminus A_H$ into A_H changes both s_H and TV by exactly t. □

Proof of Lemma 6. For $A \subseteq \bar{\mathcal{G}}$, $\psi_{\#}P_X(A) - \psi_{\#}Q_X(A) = P_X(\psi^{-1}(A)) - Q_X(\psi^{-1}(A)) \le \mathrm{TV}(P_X, Q_X)$; take the supremum over A. For KL, assume $\mathrm{KL}(P_X \| Q_X) < \infty$ (otherwise there is nothing to prove) and group strings by their parse. For each $G \in \bar{\mathcal{G}}$, since $\psi_{\#}P_X(G) = \sum_{x \in \psi^{-1}(G)} P_X(x)$ and likewise for Q_X, the log-sum inequality (Cover & Thomas, 2006) gives

$$\sum_{x \in \psi^{-1}(G)} P_X(x) \log \frac{P_X(x)}{Q_X(x)} \;\ge\; \psi_{\#}P_X(G) \log \frac{\psi_{\#}P_X(G)}{\psi_{\#}Q_X(G)}.$$

Summing over G gives $\mathrm{KL}(P_X \| Q_X) \ge \mathrm{KL}(\psi_{\#}P_X \| \psi_{\#}Q_X)$. □

Proof of Lemma 3. Because $\perp \notin A_H$ and $Q(\mathcal{G}) = 1 - q > 0$, $s_H(Q) = Q(A_H \cap \mathcal{G}) = Q(\mathcal{G})\, Q(A_H \mid \mathcal{G}) = (1 - q)\, s_H(Q_{\mathrm{v}})$. Subtracting $s_H(P) = (1 - q)s_H(P) + q\, s_H(P)$ gives the second identity. □

Three consequences are used in the paper. (i) The factor $1 - q$ is positive and common to all patterns, so $s_H(Q)$ and $s_H(Q_{\mathrm{v}})$ order the patterns identically, ties included; any rank statistic, such as Kendall's τ_a or Spearman's ρ, takes the same value under the total-map and valid-only conventions whenever defined. (ii) Since $P(\perp) = 0$, $\mathrm{TV}(P, Q) \ge |P(\{\perp\}) - Q(\{\perp\})| = q$, so the bound of Lemma 2 is never smaller than the invalid rate, however well the valid outputs match; this is one reason we report invalid mass separately from in-support error. (iii) The identity holds verbatim for finite samples: if n_{raw} attempts contain $n_{\mathrm{valid}} > 0$ strict-valid outputs, every total-map support estimate equals $n_{\mathrm{valid}}/n_{\mathrm{raw}}$ times its valid-only estimate (checked in Appendix D.3).

Proof of Lemma 7. If $s_H(P) - s_{H'}(P) > 2\varepsilon$, then $s_H(Q) \ge s_H(P) - \varepsilon > s_{H'}(P) + \varepsilon \ge s_{H'}(Q)$. Let $N = |\mathcal{H}|(|\mathcal{H}| - 1)/2$ and $\tau_a = N^{-1} \sum_{\{H, H'\}} \mathrm{sign}\big(s_H(P) - s_{H'}(P)\big)\, \mathrm{sign}\big(s_H(Q) - s_{H'}(Q)\big)$. By the first statement, each pair with $|s_H(P) - s_{H'}(P)| > 2\varepsilon$ (a fraction $1 - t_P - \pi^{+}_{2\varepsilon}$ of all pairs) contributes $+1$; pairs tied under P (fraction t_P) contribute 0; the remaining pairs (fraction $\pi^{+}_{2\varepsilon}$) contribute at least -1. Hence $\tau_a \ge (1 - t_P - \pi^{+}_{2\varepsilon}) - \pi^{+}_{2\varepsilon} = 1 - 2\pi^{+}_{2\varepsilon} - t_P$. □

Proof of Corollary 4. By linearity, $s_H(M_\lambda) - s_H(\hat{P}_{\mathrm{tr}}) = \lambda \left(s_H(R) - s_H(\hat{P}_{\mathrm{tr}})\right)$, and both supports lie in $[0, 1]$, so $|s_H(M_\lambda) - s_H(\hat{P}_{\mathrm{tr}})| \le \lambda$ for all H simultaneously. Since $\hat{P}_{\mathrm{tr}}$ puts no mass outside the training set and R puts all of its mass there, M_λ puts exactly mass λ outside the training set. The mixture can also be realized over strings, the setting of our models. Restrict R to parseable graphs: for any accepted string x_* whose graph $G_* = \psi(x_*)$ is absent from training, take R and R_X to be point masses at G_* and x_*, respectively. Mix R_X with a uniform sample of stored training rows, with weights λ and $1 - \lambda$; linearity of pushforward gives M_λ under the idealized round-trip assumption $\psi_{\#}P_X = P$ of Section 3.1. In the actual data, the parsed-training distribution and $\hat{P}_{\mathrm{tr}}$ over stored graphs differ in total variation by at most the fraction of training rows whose string does not parse back to the stored molecule (0.033% for DFS codes and 0.028% for node/edge-addition sequences; Appendix B). □

Remark 8 (Replay already matches population supports). Take P to be the empirical distribution of the whole corpus and let the n training graphs be drawn from it, i.i.d. or uniformly without replacement (our split is a uniformly random permutation of the corpus), with empirical distribution $\hat{P}_{\mathrm{tr}}$. For a pattern family $\mathcal{H}$ fixed independently of the training sample, Hoeffding's inequality, which covers sampling without replacement, and a union bound over $\mathcal{H}$ give, with probability at least $1 - \delta$, $\max_{H \in \mathcal{H}} |s_H(\hat{P}_{\mathrm{tr}}) - s_H(P)| \le \sqrt{\log(2|\mathcal{H}|/\delta)/(2n)}$. For $|\mathcal{H}| = 253$, $n = 3{,}371{,}958$ and $\delta = 0.05$ this is 0.0012: with probability at least 0.95, replaying the training set reproduces every support of such a family to within 0.0012 of P. On the same event, by Corollary 4 and the triangle inequality, a generator that replays with probability $1 - \lambda$ and emits arbitrary unseen graphs otherwise stays

within $\lambda + 0.0012$ of every support in the family. Our panel has this size but was mined from 10,000 of the training rows, so it does not meet the independence assumption: the numerical guarantee is illustrative, not a confidence bound for our selected panel. Support agreement alone cannot separate replay from the generation of new graphs.

B SERIALIZATIONS

We specify the two serializations of Section 3.2 in enough detail to reproduce them, and document two of their properties. (i) Neither string order is isomorphism-canonical, and the DFS-code parser accepts strings the serializer never writes, so several strings decode to one graph; this is why Section 3.1 defines Q as a pushforward. (ii) A small fraction of training strings does not parse back to the complete molecule; this fraction bounds how far the parsed training data depart from the idealization $\psi_{\#}P_X = P$ (proof of Corollary 4). Both serializations are computed from the same labeled graphs, and row k of either dataset is the same molecule in every split. Table 2 summarizes both.

Table 2: The two serializations of the PCQM4Mv2 training split. DFS lengths count edge tuples (without <s>/</s>); node/edge-addition lengths count all tokens including <s> and </s>. $|V|$ and $|E|$ are the numbers of atoms and bonds. In DFS training the SFT trainer appended a second </s>; sampling stops at the first </s>.

	DFS code	Node/edge-addition sequence
Token	one edge tuple $\langle i, j, \ell_i, \ell_j, \ell_{ij} \rangle$	one action
Vocabulary (incl. special tokens)	12,297 (12,292 tuples, 5 special)	86 (82 actions, 4 special)
Length: mean / median / p99 / max	15.98 / 16 / 23 / 68 tuples	60.39 / 63 / 81 / 223 tokens
Maximum length, validation / test	53 / 55 tuples	181 / 195 tokens
Length of one molecule	$2\|E\| - \|V\| + 1$ (if connected)	$2\|V\| + 2\|E\| + 3$
Tokens per pass over the training split	63,993,170	203,647,308
Model-visible framing in training	<s> ... </s> </s>	<s> ... END_NODE </s>
Encodes	start atom's connected component	all atoms and bonds
Isomorphism-invariant	no	no
Used in	Sections 4 and 6	Section 5

DFS codes. Each token is one edge tuple <i,j,ℓ_i,ℓ_j,ℓ_{ij}>: the discovery indices of the two endpoints, their element symbols, and the bond label SINGLE, DOUBLE or TRIPLE. From the start atom, a recursive depth-first traversal visits the neighbors of the current atom v in ascending stored atom index. An unvisited neighbor u emits $\langle \mathrm{idx}(v), m, \ell_v, \ell_u, \ell_{vu} \rangle$, where m is the next unused discovery index, and the traversal recurses into u; a visited neighbor that is not the parent of v emits $\langle \mathrm{idx}(v), \mathrm{idx}(u), \ell_v, \ell_u, \ell_{vu} \rangle$. There is no rightmost-path restriction.

Ring closures are emitted twice. The second rule fires at both endpoints of every ring-closing bond: first as a *closure* with $i > j$, when the later-discovered endpoint scans its ancestor, and later as a *reciprocal copy* with $i < j$, when the traversal has returned to the ancestor. Tokens carry no direction flag, so a reciprocal copy looks like a forward tuple whose j is not a new index. A single-bonded carbon triangle stored as atoms 0, 1, 2 becomes

<0,1,C,C,SINGLE><1,2,C,C,SINGLE><2,0,C,C,SINGLE><0,2,C,C,SINGLE>,

a closure followed by its reciprocal copy.

Start atom; deterministic, not canonical. The molecule is serialized from every start atom, and the string whose *first tuple* is smallest is kept (ties: earliest stored start atom). The start atom thus minimizes its element symbol, then the symbol of its lowest-index neighbor, then the bond label, with symbols compared as strings; a start atom without neighbors yields the empty string, which compares as smallest. This is not gSpan's minimum DFS code (Yan & Han, 2002), which compares whole codes, is invariant to atom numbering, and is generated by GraphGen (Goyal et al., 2020). Our string is fixed for a stored numbering, but the tie-break and neighbor order depend on it: all 120 relabelings of one five-atom labeled graph gave 24 distinct strings.

Coverage. The traversal visits only the start atom's connected component: in the training split, 580 strings of multi-fragment molecules encode one fragment, 527 strings are empty because the start atom has no neighbors, and the other 3,370,851 rows parse to a graph labeled-isomorphic to the complete molecule. The first two groups are the DFS-code cases of property (ii) above. The TUDataset runs (Appendix H) use a separate DFS implementation, a greedy rightmost-path extension that emits each backward edge once.

Node/edge-addition (DGMG-style) sequences. Each atom contributes `ADD_NODE[N`a`]` for its element a, then one pair `ADD_EDGE[E`b`] DEST[`k`]` for each bond of order $b \in \{1.0, 2.0, 3.0\}$ to an earlier atom with 0-based addition index k, then `END_EDGE`; `END_NODE` follows the last atom. Only DGMG's action format (Li et al., 2018) is used; the generator is the LLaMA-style Transformer of Appendix C, not DGMG's graph network. Atoms are added in order of (element symbol as a string, stored atom index), and a new atom's bonds are listed by bond order, then by `DEST` index. Because the order follows labels rather than connectivity, partial graphs are often disconnected: only 539,740 training strings (16.0%) attach every non-first atom to an earlier atom at insertion. The sequence encodes every atom and bond, but the strict parser rejects disconnected graphs, so the 959 multi-fragment training molecules are strict-invalid; they are the construction-sequence cases of property (ii).

C Data, Models, and Compute

Data. We load 3,746,620 PCQM4Mv2 molecular graphs (Hu et al., 2021). The random 90/5/5 split (Section 3.2) makes the validation and test partitions same-corpus samples. Vertex labels are RDKit element symbols (RDKit, 2026) and edge labels Kekulé bond orders; hydrogens are implicit: an explicit H vertex occurs only 17 times in the training graphs. Table 3 lists the statistics, including the held-out rows whose DFS string occurs in training; the held-out reference (Appendix D) removes these overlaps.

Table 3: PCQM4Mv2 as used in this paper. Statistics are recomputed from the graphs and strings actually used for training and evaluation. The split is a seed-42 uniformly random 90/5/5 partition of all molecules, not the official OGB-LSC split. A stored string is rejected by the strict parser only when it is empty (DFS) or encodes a disconnected graph (node/edge-addition).

Molecules (none dropped by either serializer)	3,746,620
Training / validation / test graphs	3,371,958 / 187,331 / 187,331
Atom labels	32 element symbols
Bond labels	single, double, triple (Kekulé)
Atoms per molecule: mean / max / min	14.14 / 51 / 1
Bonds per molecule: mean / max	14.56 / 59
Mean degree / mean density per molecule	2.05 / 0.163
Training molecules with at least one ring closure	2,971,402 (88.1%)
Training molecules with several fragments / a single atom	959 / 148
Distinct training strings: DFS / node/edge-addition	3,104,677 / 3,105,498
Validation / test rows whose DFS string occurs in training	25,291 (13.5%) / 25,557 (13.6%)
Stored DFS strings rejected by the strict parser (train / val. / test)	527 / 32 / 24
Stored node/edge-addition strings rejected (train / val. / test)	959 / 48 / 55

Models and training. All autoregressive Transformers trained on PCQM4Mv2 or TUDataset are `LlamaForCausalLM` decoders (Touvron et al., 2023) initialized at random (the DiGress denoiser of Appendix G is the exception), with RMSNorm ($\epsilon = 10^{-6}$), rotary position embeddings ($\theta = 10{,}000$), as many key/value heads as query heads, and at most 2,048 positions. Table 4 gives the remaining configuration and exact parameter counts, and Table 5 the training recipes. The headline DFS-code model of Section 4 and the target-removal models of Section 5 share the *small* body and differ in vocabulary, data, schedule and token budget. LSTM, n-gram and learning-rate-probe models are described in Appendix K, and the baselines of Section 4.3 in Appendix F.

Sampling. Sampling is unconditional: the prompt is the single token `<s>`, and a draw ends at the first `</s>` or at the length limit. For the main support measurements of Section 4 (Table 1), a

Table 4: Model configurations and exact parameter counts. All models in this table are LLaMA-architecture decoder-only Transformers trained from random initialization (SwiGLU feed-forward, RMSNorm, rotary position embeddings, head dimension 64, no biases, untied input and output embeddings). Names are not ordered by size: *medium* is smaller than *small*. Non-embedding parameters exclude the input and output embedding matrices. Total parameters are given for the DFS vocabulary (12,297 tokens) and, for the one body trained with it, the action vocabulary (86 tokens).

						Total parameters		
Name	Layers	Width	Heads	FFN	Non-embedding	DFS	action	Used in
tiny	4	256	4	1,024	4,196,608	10,492,672	—	Section 6
medium	16	576	9	2,304	84,953,664	99,119,808	—	Section 6
small	12	768	12	3,072	113,265,408	132,153,600	113,397,504	Sections 4–6
large	16	1,280	20	5,120	419,472,640	450,952,960	—	Section 6

Table 5: Training recipes. Both families minimize next-token cross-entropy from random initialization with AdamW (Loshchilov & Hutter, 2019) ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$), weight decay 0, gradient-norm clipping at 1.0, bf16 arithmetic, and gradient checkpointing, on one NVIDIA A100 80GB per run. $N = 113{,}265{,}408$ is the number of non-embedding parameters of the *small* body.

	DFS family (Sections 4 and 6)	Target-removal models (Section 5)
Serialization, model	DFS code; tiny, medium, small, large (Table 4)	node/edge-addition; small body, 113,397,504 parameters
Training graphs	3,371,958	full data 3,371,958; C_3 withheld and C_3 matched removal 3,178,084 each; P_6 withheld and P_6 matched removal 3,193,338 each
Batch	10,240 graphs per update (2,048 × 5 accumulation; large: 256 × 40), ≈193.9k non-padding tokens	consecutive graphs accumulated until ≥131,072 non-padding tokens (mean 131,102.9, max 131,197)
Learning rate	peak 10^{-5}; no warmup; cosine decay to 0 at the last update	peak 10^{-5}; linear warmup over 45,306,163 tokens; cosine decay to 0 at $40N = 4{,}530{,}616{,}320$ tokens; training stops at $20N$, at $\approx 0.51\times$ peak
Length of training	50 epochs, 16,500 updates, 3,199,658,500 tokens (large: 12 epochs, 3,960 updates, 767,918,040 tokens)	$20N = 2{,}265{,}308{,}160$ tokens, 17,279 updates; 11.12 passes over full data, 11.83 over the C_3 sets, 11.73 over the P_6 sets
Checkpoint used	final (update 16,500; large 3,960); Section 6 also uses intermediate checkpoints	$20N$ (update 17,279)
Training runs	small: headline run (seed 42) and four replicas (seeds 1, 2, 3, 42); tiny, medium, large: one run each	15 runs: seeds 1, 2, 3 for each of full data, C_3 withheld, C_3 matched removal, P_6 withheld, P_6 matched removal (P_6 reuses the full-data runs); all conditions of a seed share one initialization
Implementation	TRL `SFTTrainer` with DeepSpeed ZeRO stage 1 (Rasley et al., 2020; Rajbhandari et al., 2020); right padding, padded positions excluded from the loss; no packing	Hugging Face Trainer on one GPU without DeepSpeed; token-exact updates; fp32 master weights with bf16 autocast

generation seed (0–4) fixes the random-number generator of one sampling run of the final headline checkpoint; the five generation seeds are Monte Carlo repetitions of one trained model, not independently trained models. Each generation seed draws 4,096 raw attempts per decoder with a limit of 150 tokens; no draw under pure sampling, top-k 50 alone, or top-k 50 with top-p 0.95 reached it, so under these decoders no output is invalid because it was cut off. Table 7 lists the six decoders, and Appendix I.4 specifies target-removal sampling.

Software and compute. Pattern mining uses a patched copy of the gBolt implementation of gSpan (Zhou, 2019; Yan & Han, 2002); containment and isomorphism tests use the VF2 matcher of NetworkX (Cordella et al., 2004; Hagberg et al., 2008). Every Transformer and LSTM training and generation job used one NVIDIA A100 80GB GPU; DiGress (Appendix G) was trained on an NVIDIA GB10, and the n-gram, k-gram and bridge baselines ran on CPU. The reference-support run (mining, and exact VF2 counting of the three panels on the reference and audit sets) took 30,945 s of wall-clock time on 24 workers; training compute is given in the Reproducibility Statement.

D Measurement Details

The second panel is described in Appendix E.3 and the re-mined protocol in Appendix E.

D.1 Strict parsers and novelty

Validity decides which outputs enter the valid-only supports and which count as invalid mass q (Section 4.2). Exact-string novelty decides which outputs enter the string-novel rows (Section 4.1); isomorphism novelty decides which outputs, including target-removal hits, reproduce no training graph (Sections 4.1 and 5). A *strict* parser either reads the whole string into a labeled graph or returns $\perp$; it never repairs, skips or reorders graph tokens, so every counted graph is exactly the graph the string encodes.

DFS codes. After removing `<pad>`, `<s>`, `</s>` and whitespace, a string is strict-valid if and only if: (i) it is a nonempty concatenation of tuples `<int,int,label,label,label>` with no leftover characters; (ii) the first tuple is $(0, 1)$; (iii) no tuple is a self-loop; (iv) every index not seen before is exactly the next unused index; (v) a repeated index carries the same atom label as before; (vi) no directed pair (u, v) occurs twice, and a tuple (v, u) for an existing bond (u, v) is accepted only if its bond label agrees, in which case it adds nothing to the graph; (vii) the resulting graph is connected. Reciprocal copies are not required, depth-first order is not enforced, and valence is not checked.

Node/edge-addition sequences. A string must have exactly one leading `<s>`, one final `</s>`, and no other special or padding token. A state machine then reads the actions: `ADD_NODE` only when the previous atom is closed; `ADD_EDGE` only for an open atom with no pending bond; `DEST[`k`]` only for a pending bond, naming an earlier atom; no bond twice; `END_EDGE` only with no pending bond; `END_NODE` only when every atom is closed, with nothing after it; unknown tokens are invalid. The string must terminate with at least one atom, and a graph with more than one atom must be connected.

Structural, not chemical. Strict validity is a statement about graph syntax and connectivity only. RDKit sanitization (RDKit, 2026) is a diagnostic for target-removal outputs and never filters or repairs an output; the bridge baseline does apply an RDKit filter (Appendix F).

Novelty. Exact-string novelty compares the normalized string (special tokens and whitespace removed) with all 3,371,958 normalized training strings; duplicates among generated outputs are kept. Isomorphism novelty is tested against the unfiltered training split, including every graph that a removal condition deleted, so an output that reproduces a withheld molecule is not novel. Graphs are keyed by exact isomorphism invariants (label counts, labeled degree multisets, a four-iteration Weisfeiler–Leman hash; Shervashidze et al., 2011), and exact VF2 labeled isomorphism is tested only within a key, so the keying never changes a verdict.

D.2 Pattern panel, reference supports and metrics

Within a panel, the reference and metrics are the same for every generated and real-data set of Table 1, so its rows are measured identically; they differ in which outputs are kept, and the held-out row also carries the train/test difference quantified below.

Panel. The 10,000 mining rows (9,998 strict-valid) were drawn uniformly without replacement with a fixed seed; relative minimum support 0.1 is an absolute threshold of 999 graphs.

Reference supports and containment. In each empirical comparison, P is the uniform distribution over the specified reference rows, not a population distribution (Section 3.3). The references are (a) 200,000 strict-valid training graphs drawn without replacement with a fixed seed, and (b) 161,352 held-out test graphs: the 187,307 strict-valid test rows minus 25,533 rows whose string occurs in training and 422 duplicates. The two references differ by $\overline{|\Delta s|}$ 0.0070 and τ_a 0.935, a train/test difference that any comparison against (b) inherits. Containment uses NetworkX `subgraph_is_monomorphic` with categorical matching of atom and bond labels. To check that exact VF2 counting and gBolt's mining count the same support, we compared them while re-mining the 678-graph sets of Table 8 at several candidate-side minimum supports, against reference sets of 262 to 282 patterns mined at 0.1: gBolt-reported support equals the VF2 count in all 180 comparisons. The counted supports of Table 1 therefore use the containment predicate by which gSpan selected the panel.

Metrics. The mass-weighted error is $\sum_H s_H(P)|\Delta s_H| / \sum_H s_H(P)$. Kendall's $\tau_a = (C-D)/N$ runs over all $N = 31{,}878$ pairs in the 253-pattern panel, with C concordant and D discordant pairs; tied pairs count as neither. Reported dispersion is the standard deviation across generation or sampling seeds, not a confidence interval for the mean. For the Transformer at the matched size $n = 1{,}024$ we draw 200 uniform subsamples without replacement from each generation seed's strict-valid (or strict-valid and exact-string-novel) outputs; $\pm$ is the standard deviation of a single draw.

D.3 Validity split and sampling floors

Averaging consequence (iii) of Appendix A over the panel gives, for the signed mean, $\overline{\Delta s}_{\text{total}} = (1-\hat{q})\,\overline{\Delta s}_{\text{valid}} - \hat{q}\,\bar{s}_P$, with $\hat{q}$ the observed invalid fraction and $\bar{s}_P = 0.2214$ the panel mean of the reference supports. In all 75 evaluated sets (Transformer pools and bridge and n-gram baseline sets) this agrees with the stored values up to floating-point rounding, so invalid outputs enter the total map as empty rows. On the headline model's full pure-sampling pools ($\hat{q} = 0.1416$), the total-map mass-weighted error is 0.0434 ± 0.0010, against 0.0067 ± 0.0013 for valid-only. To see what total-map error invalidity alone would produce, we add 580 empty graphs to 3,516 real graphs, the mean numbers of invalid and strict-valid outputs, respectively, among a generation seed's 4,096 attempts; this gives a total-map mass-weighted error of 0.0453 with training graphs from (a) and 0.0429 with held-out graphs (200 Monte Carlo repetitions). The Transformer's 0.0434 lies between these two reference scores, consistent with invalidity accounting for most of the total-map error. This comparison concerns absolute error; the signed-mean identity above does not give an additive decomposition of absolute error.

Section 4.1 uses the i.i.d. floor of Table 6 as a second benchmark beside same-size real resamples. One real-data row falls below it: at $n = 1{,}024$ the five training resamples of Table 1 average $\overline{|\Delta s|} = 0.0083$, against the floor's 0.0095. This reference expectation is not a lower bound on an individual draw; with single-draw SD 0.0015, the difference is compatible with sampling variation.

Table 6: Ideal sampling floor on the 253-pattern panel: the expected error of a sampler that draws n graphs i.i.d. from the reference distribution, $\mathbb{E}\,|\mathrm{Bin}(n, s_H(P))/n - s_H(P)|$ averaged over the panel (unweighted and weighted by $s_H(P)$), with the supports of the 200,000-graph training reference treated as exact. This is an i.i.d. sampling benchmark, not a universal lower bound on error. The subsampling check draws n rows without replacement from that reference itself (in-sample). *Means under pure ancestral sampling.

	Analytic floor		Subsampling check			
n	$\overline{\lvert\Delta s\rvert}$	mass-wtd.	$\overline{\lvert\Delta s\rvert}$	mass-wtd.	τ_a	Where this n occurs
1,024	0.0095	0.0102	0.0094	0.0100	0.899	matched-n comparison (Table 1)
2,652	0.0059	0.0063	—	—	—	string-novel outputs per generation seed*
3,516	0.0051	0.0055	—	—	—	strict-valid outputs per generation seed*
4,096	0.0048	0.0051	0.0047	0.0051	0.947	raw attempts per generation seed

Table 7: Support reproduction by decoder on full pools (PCQM4Mv2, DFS codes; 253-pattern panel; reference supports from 200,000 training graphs; exact counting). One trained model; five generation seeds of 4,096 attempts per decoder; means over seeds, ± their SD where shown. Validity: share of attempts accepted by the strict parser. n_{v}: strict-valid graphs per seed. Novel: share of these that are not a training string. Atoms: their mean size. $\overline{|\Delta s|}$, signed $\overline{\Delta s}$, τ_a and ρ are valid-only (τ_a and ρ are identical on the total map). Total mw: mass-weighted mean $|\Delta s|$ on the total map. Resamples: 4,096 training or held-out test graphs (five seeds; all valid). Errors shrink roughly as $1/\sqrt{n}$, so rows with different n_{v} are not directly comparable; Table 1 compares at $n = 1{,}024$.

Decoder	Validity	n_{v}	Novel	Atoms	$\overline{\lvert\Delta s\rvert}$	$\overline{\Delta s}$	τ_a	ρ	Total mw
Pure sampling	0.858	3,516	0.754	14.10	0.0062 ± 0.0011	$+0.0010$	0.931	0.992	0.0434
Top-k 50	0.941	3,853	0.765	14.13	0.0057 ± 0.0009	$+0.0004$	0.936	0.993	0.0180
Top-p 0.95	0.968	3,964	0.700	14.67	0.0249	$+0.0235$	0.895	0.980	0.0198
Top-k 50, top-p 0.95	0.983	4,026	0.685	14.72	0.0256 ± 0.0010	$+0.0241$	0.892	0.979	0.0244
T=1.2, top-p 0.98	0.639	2,617	0.824	13.24	0.0272	-0.0247	0.822	0.950	0.1333
T=0.8, top-p 0.90	0.995	4,076	0.478	15.43	0.0685	$+0.0429$	0.611	0.784	0.0732
Resample: training	1	4,096	—	14.16	0.0046 ± 0.0009	-0.0001	0.949	0.995	0.0050
Resample: held-out	1	4,096	—	14.31	0.0081 ± 0.0019	$+0.0023$	0.923	0.989	0.0096

E Additional Support Results

This appendix gives further results behind Section 4, where "the Transformer" is the headline DFS-code model: one training run, whose generation seeds are sampling streams of that one model.

Table 7 holds the trained model fixed and varies only the decoding rule, giving the full-pool numbers behind Section 4.2. Under pure sampling, the valid-only error (0.0062 ± 0.0011 at 3,516 graphs per seed) and string-novel error (0.0083 ± 0.0005 at 2,652) exceed their own-n i.i.d. floors (0.0051 and 0.0059; Table 6) by about 0.001 and 0.0023. Top-p 0.95 over-represents (valid-only) 96.0% (alone) and 96.3% (with top-k 50) of the 253 patterns, against 53.3% (pure) and 50.5% (top-k only). Counted valid-only, pure sampling thus errs in both directions, whereas top-p truncation pushes almost every pattern upward: a decoder bias that ρ barely registers (Section 4.2).

The strata, isomorphism and replica checks of Section 4.1 score sets with the *re-mined protocol* of Tables 8–11 rather than by exact counting, and we use them only for rank statistics. This protocol mines reference patterns with gSpan (Yan & Han, 2002; Zhou, 2019) at minimum support 0.1 (2–8 atoms), mines each candidate set at the same relative threshold and assigns support 0 to unmined reference patterns. These zero-filled patterns are concentrated at the low-support end and, for the n-gram sets, reach into the torso (Appendix E.1). The sets of Tables 8, 9(b) and 10 were selected with a strict parser that also rejected atom labels containing a lowercase "s"; with the corrected parser, the re-mined ρ of every generated 678-graph set against the training reference changes by at most 0.0002 (held-out resample: 0.950 to 0.944), but the 678-graph columns of Table 10 depend on it. The Transformer's 678-graph sets come from a separate generation run.

E.1 Censoring and Support Strata in Re-Mined Comparisons

Section 3.3 counts generated sets exactly instead of re-mining them. Table 8 scores the same sets both ways to show why.

gSpan and exact VF2 counting agree on every support, so in Table 8 only the candidate-side threshold differs. Re-mining barely changes ρ (differences ≤ 0.004) but inflates $\overline{|\Delta s|}$ unequally: from 0.0189 to 0.0274 for bridge recombination and from 0.0165 to 0.0232 for the held-out resample, but only from 0.0312 to 0.0331 for the string-novel Transformer set. A pattern near the threshold is dropped at random, and a set biased upward (here the Transformer under top-p truncation, signed error $+0.028$) crosses the threshold more often, so re-mining rewards upward bias: the Transformer sets' lower zero-filled mass reflects over-representation, not higher fidelity. The bias also makes their exact-count error exceed the bridge set's here; under pure sampling on the fixed panel at a matched 1,024 graphs, the string-novel Transformer error is 0.0112 against 0.0138 for bridge (Table 1).

Table 8: Candidate sets scored with re-mined and with exactly counted supports (PCQM4Mv2, DFS codes). Each set has 678 strict-valid graphs; the reference is 678 random strict-valid training graphs, redrawn for each of five paired generation/reference seeds, with 262–282 reference patterns. Transformer: temperature 1, top-k 50, top-p 0.95. Re-mined: mean number per seed of reference patterns set to 0 ("zero-filled") and the share of reference support mass they carry. Exact: VF2 counting in every candidate graph; no pattern has zero support in any set. Means over the five seed pairs, $\pm$ their SD. Held-out resample: validation and test graphs.

	Re-mined candidate support			Exact candidate support			
Candidate set	Zero-filled	Mass	ρ	ρ	τ_a	$\overline{\lvert\Delta s\rvert}$	$\overline{\Delta s}$
Resample: training	29.6	0.055	0.943	0.943	0.809	0.0177 ± 0.0026	-0.0056
Resample: held-out	21.2	0.038	0.950	0.950	0.824	0.0165 ± 0.0025	$+0.0032$
Transformer, all strict-valid	12.0	0.022	0.942	0.941	0.811	0.0283 ± 0.0043	$+0.0231$
Transformer, string-novel	6.8	0.013	0.946	0.946	0.819	0.0312 ± 0.0031	$+0.0279$
Bridge recombination	27.2	0.050	0.934	0.934	0.794	0.0189 ± 0.0021	-0.0009
5-gram (exact counts)	44.6	0.087	0.864	0.866	0.700	0.0314 ± 0.0032	-0.0027
3-gram (exact counts)	69.4	0.145	0.786	0.790	0.615	0.0465 ± 0.0034	-0.0100

Table 9: Spearman ρ within support strata of the reference patterns (PCQM4Mv2, DFS codes; re-mined protocol). Head / tail: reference support at or above the 90th / at or below the 10th percentile of the reference patterns; torso: the rest. Mining uses nominal minimum support 0.1, so the tail is the lowest-support tenth of *frequent* patterns, not rare substructures. (a) Lenient parser (a regular-expression tuple extractor that never rejects an output, so strict-invalid outputs enter as graphs); 253 reference patterns (26 head, 201 torso, 26 tail; tail band 0.1002–0.1127). Transformer: temperature 1, top-k 50, top-p 0.95, three draws with an unseeded random-number generator (mean $\pm$ SD). Real-data resample: 10 bootstrap samples of 1,024 strings from the reference's own unique strings. (b) The sets of Table 8 (mean $\pm$ SD over five seed pairs); tail band 0.0988 to 0.103–0.108.

Candidate set	Head ρ	Torso ρ	Tail ρ
(a) Lenient parser, $n = 1{,}024$, reference: 10,000-string training subsample			
Transformer, three repeated draws	0.984 ± 0.002	0.946 ± 0.005	0.471 ± 0.151
Real-data resample (in-sample)	0.987 ± 0.006	0.969 ± 0.008	0.288 ± 0.165
(b) Strict parser, $n = 678$, reference of 678 training graphs			
Resample: training	0.975 ± 0.016	0.922 ± 0.023	-0.008 ± 0.220
Resample: held-out	0.975 ± 0.013	0.929 ± 0.024	0.227 ± 0.202
Transformer, all strict-valid	0.967 ± 0.007	0.924 ± 0.021	0.039 ± 0.149
Transformer, not a training string	0.968 ± 0.009	0.930 ± 0.017	0.142 ± 0.200
Bridge recombination	0.965 ± 0.012	0.912 ± 0.018	0.126 ± 0.089
5-gram (exact counts)	0.943 ± 0.016	0.828 ± 0.029	0.142 ± 0.136
3-gram (exact counts)	0.894 ± 0.020	0.728 ± 0.019	0.128 ± 0.195

Section 4.1 states that rank order among the lowest-support panel patterns is unresolved even for real-data resamples; Table 9 shows this within support strata. Its tail is a narrow band of reference support, and near support 0.105 the binomial standard error is 0.0096 at $n = 1{,}024$ and 0.0118 at $n = 678$, comparable to or larger than the band width. Tail agreement is therefore low and unstable for every set, including real-data resamples, and the Transformer's tail values fall within their spread: this measurement cannot resolve ranking within the band.

E.2 Held-Out References, Isomorphism Novelty and Replicas

These checks address two ways the agreement of Section 4.1 could mislead: string-novel outputs may be training graphs written as another DFS string, and the headline model is a single training run. Scoring against held-out rather than training references also tests agreement with molecules outside the training split that the Transformer and the bridge baseline draw on. Both checks use re-mined ρ (temperature 1, top-k 50, top-p 0.95 unless stated), which is not comparable with Table 1.

Table 10: Re-mined Spearman ρ against held-out references (validation or test partition of the same random PCQM4Mv2 split; not an out-of-distribution test); mean $\pm$ SD over five paired generation/reference seeds. Transformer: temperature 1, top-k 50, top-p 0.95. Left: 2,597 graphs per set, the fewest isomorphism-novel Transformer outputs among five generation seeds of 4,096 attempts (bridge: separate run, 4,096 per seed). Right: 678 per set. Both parts use the parser variant of Appendix E. Isomorphism-novel: not labeled-isomorphic to any training graph. —: not computed.

	$n = 2{,}597$		$n = 678$	
Candidate set	Validation ref.	Test ref.	Validation ref.	Test ref.
Resample: held-out	0.985 ± 0.006	0.985 ± 0.003	0.949 ± 0.017	0.952 ± 0.014
Resample: training	0.980 ± 0.008	0.977 ± 0.007	0.939 ± 0.016	0.946 ± 0.014
Transformer, all strict-valid	—	—	0.930 ± 0.032	0.936 ± 0.013
Transformer, string-novel	0.965 ± 0.010	0.965 ± 0.006	0.923 ± 0.030	0.930 ± 0.014
Transformer, isomorphism-novel	0.966 ± 0.009	0.966 ± 0.003	—	—
Bridge, string-novel	0.961 ± 0.008	0.958 ± 0.012	0.910 ± 0.021	0.924 ± 0.013
Bridge, isomorphism-novel	0.959 ± 0.009	0.958 ± 0.010	—	—

Table 11: Spearman ρ across independently trained models (PCQM4Mv2, DFS codes; re-mined protocol): 700 strict-valid outputs per generation seed against one fixed set of 700 strict-valid training graphs. Headline: mean $\pm$ SD over five generation seeds. Seed s: a model retrained from scratch with training seed s and the headline recipe (not the headline model), mean over ten generation seeds (shown for two decoders). Replicas: grand mean over the four models $\times$ ten generation seeds.

Decoder	Headline	Seed 1	Seed 2	Seed 3	Seed 42	Replicas
Top-k 50, top-p 0.95	0.950 ± 0.015	0.952	0.946	0.945	0.952	0.949
Top-p 0.95	0.947 ± 0.007					0.950
Top-k 50	0.947 ± 0.013					0.954
Pure	0.956 ± 0.009	0.958	0.953	0.951	0.953	0.954
T=1.2, top-p 0.98	0.926 ± 0.022					0.930
T=0.8, top-p 0.90	0.785 ± 0.007					0.774

At 2,597 graphs per set (Table 10), removing Transformer outputs isomorphic to a training graph leaves ρ unchanged (0.966 against 0.965 for string-novel outputs, for both references), and the same holds for bridge recombination. Both generators fall below a held-out resample (0.985): the paired difference between isomorphism-novel Transformer outputs and the held-out resample is -0.019 for both references ($p = 0.016$ validation; $p = 0.0002$ test), whereas that between isomorphism-novel Transformer and bridge outputs is $+0.006$ ($p = 0.29$) and $+0.007$ ($p = 0.23$). Training graphs written as another string therefore do not drive the agreement; the gap to a held-out resample is detectable at this n, whereas the difference from bridge recombination is not. With this decoder (five generation seeds of 4,096 attempts), exact-string replay is 0.309 of attempts, isomorphic replay 0.3445 and string-novel strict-valid outputs 0.674, so about 5.2% of string-novel strict-valid outputs are training graphs written as a different DFS string (bridge children, separate run: about 23%).

Four further runs of the same recipe test whether re-mined ρ depends on the training run. For these independently trained replicas with temperature 1, top-k 50, top-p 0.95, the per-model mean ρ is 0.945–0.952 (Table 11). A random-effects model (training run random, generation seed a fixed block) gives a between-run SD of 0.0024 against a residual SD of 0.0104 (F-test $p = 0.224$); the 95% upper bound of the ratio of the two SDs is 1.44, so the test does not establish equivalence, only that between-run variation is not detectable against generation-seed variation at this n.

E.3 A Second Frozen Panel (Minimum Support 0.01)

The 253-pattern panel covers only frequent patterns (Section 3.3), so agreement on it says nothing about rarer substructures. This panel, mined after the 253-pattern results from the same 10,000 molecules at a tenfold lower threshold, is a post hoc check of whether the 253-pattern result of Section 4.1 extends to patterns down to about 1% reference support.

Table 12: Support reproduction on the 3,754-pattern panel (minimum support 0.01) by stratum of reference support $s_H(P)$ (PCQM4Mv2, DFS codes; 200,000-graph training reference; exact counting; $n = 1{,}024$ graphs per set; the sets and matched draws of Table 1). Strata are half-open intervals; entries are $\overline{|\Delta s|}$ and Kendall τ_a within the stratum, averaged over five resampling or generation seeds (Transformer: one trained model, pure ancestral sampling, 200 subsamples per seed). i.i.d. floor: expected $\overline{|\Delta s|}$ of 1,024 i.i.d. draws (exact binomial) and mean τ_a of 200 random 1,024-graph subsets of the reference graphs themselves.

Reference support	< 0.02		0.02–0.05		0.05–0.1		≥ 0.1	
Patterns (cyclic)	1,946 (86)		1,137 (38)		417 (13)		254 (5)	
Source	$\overline{\lvert\Delta s\rvert}$	τ_a	$\overline{\lvert\Delta s\rvert}$	τ_a	$\overline{\lvert\Delta s\rvert}$	τ_a	$\overline{\lvert\Delta s\rvert}$	τ_a
Resample: training	0.0028	0.446	0.0041	0.647	0.0060	0.698	0.0082	0.908
Resample: held-out	0.0030	0.433	0.0046	0.605	0.0072	0.669	0.0120	0.886
Transformer, strict-valid	0.0030	0.432	0.0045	0.613	0.0066	0.659	0.0101	0.892
Transformer, string-novel	0.0031	0.419	0.0048	0.598	0.0075	0.633	0.0113	0.880
Bridge baseline	0.0037	0.350	0.0055	0.576	0.0080	0.612	0.0139	0.853
5-gram	0.0052	0.291	0.0094	0.419	0.0196	0.334	0.0257	0.756
3-gram	0.0075	0.193	0.0143	0.276	0.0295	0.207	0.0456	0.635
i.i.d. floor ($n = 1{,}024$)	0.0029	0.448	0.0043	0.626	0.0063	0.685	0.0095	0.898

This panel, shown in Figure 1 and the last two columns of Table 1, has 3,754 patterns (142 cyclic) of 2–8 atoms, is counted exactly on the same reference, sets and matched draws as the 253-pattern panel, and contains, as identical entries, the 253-, 672- and 1,825-pattern panels mined from the same molecules at minimum support 0.1, 0.05 and 0.02.

Over the whole panel (Table 1), the strict-valid Transformer set reaches $\overline{|\Delta s|} = 0.0043$ against an i.i.d. floor of 0.0041 (expected error of 1,024 i.i.d. draws), and the order of the rows is that of the 253-pattern panel, except that the held-out resample and the string-novel Transformer outputs swap in $\overline{|\Delta s|}$. Kendall τ_a is lower for every row than on the 253-pattern panel because rarer, closely spaced supports are harder to rank from 1,024 graphs: an ideal sampler reaches only $\tau_a = 0.786$ here (0.899 on the 253-pattern panel). Within each stratum (Table 12), the strict-valid Transformer's error stays close to the i.i.d. floor, while both n-gram models fall furthest below the floor's τ_a in the 0.05–0.1 stratum (0.334 and 0.207 against 0.685). The Transformer's whole-panel and within-stratum errors thus extend the agreement of Section 4.1, at matched n, to patterns down to about 1% reference support. Bridge recombination leaves 38–55 patterns per seed with zero support (55 in the union, all with an N–O bond, 2 cyclic; Appendix F), whereas the Transformer's full pure-sampling strict-valid and string-novel pools (five generation seeds of 4,096 attempts) miss none.

F Baseline Generators

Section 4.3 asks whether simple non-neural generators reach the Transformer's agreement; one that does illustrates that agreement alone does not identify how graphs are generated (Corollary 4). We use two: bridge recombination and exact-count n-grams.

The baselines produce labeled graphs or DFS strings that we score exactly like Transformer outputs; their seeds (0–4) are generation seeds. Both are rejection-sampled: each seed collects 1,024 accepted outputs, all strict-valid and not training strings, and rejected attempts are discarded rather than counted as invalid outputs. For these sets the total, valid-only and string-novel conventions therefore coincide.

Bridge fragment recombination. This is the specified bridge-only baseline of Sections 4.3 and 5.3. (1) Draw two distinct rows uniformly from the 3,371,958-row training split. (2) In each parent, choose a bridge bond (one whose removal disconnects the molecule) such that both bonds carry the same bond label and every side has at least two atoms; the pair is chosen uniformly among compatible pairs. (3) Exchange the fragments and reconnect each exchanged pair with a bond of the same label, giving two complementary children per attempt. (4) Keep a child only if it is connected,

passes RDKit sanitization (RDKit, 2026) on the charge-free label graph, is not a training string, and is new among the accepted children. Each child consists of two fragments of training graphs joined by one new bond, which is a bridge of the child; applied to any training set, the procedure is therefore an instance of the construction class of Lemma 5.

Two caveats apply. First, the RDKit filter builds neutral atoms from element labels, so a nitro nitrogen (valence 4) fails sanitization; this removes nitro-containing children and depletes N–O motifs generally. On the 3,754-pattern panel (Appendix E.3), `ON=O`, `CN=O` and `CN(O)=O` (in SMILES-like notation) have zero support in all five bridge seeds; on the 253-pattern panel no generator misses a pattern. These zeros are therefore not evidence about what fragment recombination can produce, nor about rings. Second, about 23% of string-novel children are isomorphic to a training graph (Appendix E.2).

Exact-count n-gram models. These are finite-context n-gram models over DFS edge tokens of order $k \in \{3, 5\}$. A state is a whole DFS edge tuple ($\langle i, j, \ell_i, \ell_j, \ell_{ij} \rangle$) or a begin or end symbol; the context is the previous $k-1$ tuples, padded with begin symbols. Transition probabilities are exact maximum-likelihood counts over all 3,371,431 non-empty training strings, with no smoothing and no back-off. Sampling is ancestral from the begin symbol to the end symbol, capped at 512 tuples. An output is accepted if it passes the strict parser, is not a training string, and is new among accepted outputs. They test whether short token contexts alone reach the Transformer's agreement. On the panel both models under-produce the five ring-containing patterns (2.06% of reference support): at $n = 1{,}024$ the mean signed error over these patterns is -0.059 for the 5-gram and -0.148 for the 3-gram, against $+0.008$ for bridge recombination (five seeds; 200,000-graph training reference). The smoothed n-gram models of Section 6 are different models, sampled without rejection (Appendix K).

G DiGress Comparison

Table 13: Auxiliary comparison with DiGress on a 50,000-molecule PCQM4Mv2 subset (not the main training split). Reference: 39,990 strict-valid training graphs of the subset; 254 reference patterns (gSpan, absolute minimum support 3,999, 2–8 atoms). Every set is scored on its first 138 strict-valid graphs in stored order; candidate patterns are mined at absolute minimum support 13 and unmined reference patterns are set to 0. ρ: Spearman correlation over the 254 reference patterns. Patterns found: reference patterns also mined from the candidate set. Isomorphic: outputs labeled-isomorphic to a reference graph. Autoregressive comparator: a separate decoder trained on DFS codes of the same molecules (learning rate 10^{-4}, early stopping at epoch 4), sampled with temperature 1, top-k 50, top-p 0.95. Reference subsets: six sets of 198 reference graphs; mean $\pm$ SD over the six sets (per-set ρ 0.877, 0.898, 0.924, 0.929, 0.910, 0.889). One output archive per model; generation seeds unknown.

Set	Strict-valid / raw	n scored	ρ	Patterns found	Isomorphic
DiGress (archived final outputs)	200 / 200	138	0.886	233 / 254	1 / 138
Autoregressive, DFS codes	138 / 198	138	0.881	237 / 254	3 / 138
Reference subsets ($\times 6$)	198 / 198	138	0.905 ± 0.020	—	138 / 138

This comparison asks only whether support-rank agreement is specific to autoregressive sequence models; like Appendix H, it illustrates Corollary 4 and does not test how graphs are generated. DiGress (Vignac et al., 2023) was trained on a seed-42 random subsample of 50,000 PCQM4Mv2 molecules with an 80/10/10 split, not on the main training split, and the autoregressive comparator of Table 13 has the decoder architecture of Section 3.2. We rescore archived outputs of both under the re-mined protocol, not the fixed panel; strict validity is structural, not chemical. Both models lie inside the range of the six reference subsets: at $n = 138$, support-rank agreement is not exclusive to autoregressive sequence models. This does not indicate a shared mechanism: DiGress uses a Transformer denoiser, so it is not a non-Transformer control, and the values are not comparable in magnitude with Table 1.

Limitations. Three limitations apply. (i) The DiGress training set has 39,996 graphs and the support reference 39,990, so the two sets may not be identical. (ii) We use 200 final-stage DiGress

samples from one output archive of unknown generation seed. (iii) The autoregressive comparator generated 1,024 samples, but the scored set is a 198-row subset whose selection rule is unknown. With one output archive per model, the comparison does not rank the two families.

H TUDataset Benchmarks

Like Appendix G, these runs illustrate Corollary 4: on small corpora that are used entirely for training, rank agreement and exact replay coincide almost completely.

Setup. We use five TUDataset benchmarks (Morris et al., 2020): the small-molecule graphs MUTAG (Debnath et al., 1991), PTC_MR (Helma et al., 2001) and NCI1 (Wale et al., 2008), and the protein-structure graphs ENZYMES (Schomburg et al., 2004; Borgwardt et al., 2005) and PROTEINS (Dobson & Doig, 2003; Borgwardt et al., 2005). With no held-out split, the support reference is the full training set, so TU ρ measures reproduction of training-set support, not held-out generalization. Each graph is serialized both as a DFS code and as a DGMG-style action sequence; each model (Table 14) is trained on all graphs for about 80M tokens and generates 1,024 samples per decoder, with the six decoders of Table 7. Support uses the re-mined protocol. Outputs are parsed leniently; strict validity is not a meaningful quality measure on TU, because the strict parser rejects disconnected graphs, which the training sets themselves contain. The generation random-number generator was not guaranteed to be seeded for the seed-42 and seed-1337 draws; the three per-seed values in Table 14 differ by training seed, and sampling noise is not controlled separately.

Rank agreement and replay. At $T = 1$, top-k 50, top-p 0.95, mean ρ over the three training seeds is 0.95–0.99 for every dataset and serialization except ENZYMES DFS codes (0.931), where gBolt's default projection cap affects the values; the lower ends of the ENZYMES and PROTEINS DFS six-decoder ranges also partly reflect the cap (Table 14). This agreement coincides with almost complete exact replay: every one of the 1,024 samples is an exact training string in 18 of the 30 model–seed runs (Novel $= 0$ in Table 14). In a separate draw per seed-42 model, labeled isomorphism adds no replay beyond string equality in any pair. Rank agreement restricted to non-replayed outputs rests on at most 57 distinct strings per model, so we do not report it.

What TU shows. With generation dominated by replay of training strings, rank agreement on TU cannot separate replay from reproduction, and TU offers nothing to remove in most runs. The same model family replays far less on PCQM4Mv2 (Appendix E.2), which is consistent with the larger corpus but not a controlled comparison. The exact-string exclusion and the target-removal tests are carried out only on PCQM4Mv2.

Table 14: TUDataset benchmarks (all graphs used for training; no held-out split). One model per dataset, serialization and training seed (42 / 1337 / 2024): 12 layers, width 768, 113–141M parameters depending on the vocabulary. DFS: deterministic DFS edge-tuple codes; DGMG: node/edge-addition action sequences (DGMG-style). ρ: Spearman correlation between training-set support and generated support over gSpan patterns mined from the training set (minimum support 0.1, 2–8 vertices; generated side mined at the same threshold, unmined patterns set to 0), for one draw of 1,024 samples per model with temperature 1, top-k 50, top-p 0.95. Six-decoder range: minimum and maximum over six decoders and three training seeds. Novel: distinct generated strings that are not training strings, per 1,024 samples and training seed; 0 means that every sample is an exact training string. $|\mathcal{H}|$: number of reference patterns, which differs between the two serializations' evaluation paths. All entries use the default gBolt projection cap (5×10^5). †The cap affects these values. For the seed-1337 checkpoint, re-mining the same samples without the cap (ENZYMES) or with the cap raised to 10^7 (PROTEINS) gives 0.9812 (capped 0.9220) and 0.9855 (capped 0.9852) at the default decoder, and 0.8715 (0.7185) and 0.8722 (0.7868) at T=0.8, top-p 0.90. ‡Never re-mined without the cap; the effect of the cap is unknown.

Dataset (graphs)	Serial.	ρ (seed 42 / 1337 / 2024)	Mean ρ	Six-decoder range	Novel	$\lvert\mathcal{H}\rvert$
MUTAG (188)	DFS	0.9830 / 0.9807 / 0.9904	0.9847	[0.942, 0.996]	0 / 0 / 0	634
	DGMG	0.9900 / 0.9821 / 0.9930	0.9883	[0.948, 0.996]	0 / 0 / 0	634
PTC_MR (344)	DFS	0.9857 / 0.9905 / 0.9855	0.9873	[0.929, 0.991]	0 / 0 / 0	127
	DGMG	0.9742 / 0.9587 / 0.9651	0.9660	[0.917, 0.991]	0 / 0 / 0	127
ENZYMES (600)	DFS†	0.9387 / 0.9220 / 0.9328	0.9312	[0.719, 0.972]	0 / 0 / 0	95,335
	DGMG‡	0.9435 / 0.9523 / 0.9573	0.9510	[0.892, 0.972]	2 / 2 / 3	96,690
PROTEINS (1,113)	DFS†	0.9797 / 0.9852 / 0.9851	0.9833	[0.781, 0.989]	57 / 57 / 55	70,257
	DGMG‡	0.9847 / 0.9857 / 0.9869	0.9858	[0.931, 0.991]	54 / 48 / 47	71,482
NCI1 (4,110)	DFS	0.9885 / 0.9896 / 0.9925	0.9902	[0.924, 0.995]	0 / 0 / 0	818
	DGMG	0.9894 / 0.9902 / 0.9900	0.9898	[0.909, 0.993]	19 / 11 / 17	832

I Target-Removal Experiments: Details

Rates and intervals are per trained model; we never pool training seeds for inference, and totals over the three seeds are descriptive.

I.1 Targets, Matching, and Selection

Section 5 relies on three properties of the targets: each is defined exactly, so that its absence from training can be verified; P_6 but not C_3 can be built by bridge-only recombination of retained training graphs, which lack the target (Appendix I.9); and neither was chosen after seeing outcomes.

Data. All experiments use the 3,371,958 training graphs of our PCQM4Mv2 split and the construction-sequence serialization of Section 3.2 (Appendix B). Its atom order is not isomorphism-canonical, and all target-removal results are specific to this serialization.

Matching semantics. Containment is graph-level and existential, as for support (Definition 1): a graph contains a target if a label-preserving subgraph monomorphism exists. C_3 is three carbon atoms pairwise joined by single bonds; we search for 3-cycles in the subgraph formed by carbon atoms and C–C single bonds. P_6 is a simple path of six carbon atoms joined by five single bonds, each of which is a bridge of the whole host graph (an edge whose deletion disconnects it); bridges are computed on the whole labeled graph before any restriction to carbon atoms or single bonds. A chord would place path edges on a cycle, so every P_6 match is an induced path.

Selection rule. Both targets were chosen by frequency, not by any model outcome. C_3 was chosen before any model was trained, among all-carbon, single-bond simple cycles of 3–8 atoms: using the first 30,000 training graphs, the rule kept lengths with frequency in $[0.02, 0.10]$ and chose the one closest to 0.05 (pilot support 1,604 of 30,000). P_6 was chosen before any C_3 or P_6 outcome was inspected (Appendix I.5), among all-carbon, single-bond paths of 4–8 atoms whose edges are all

bridges, as the length whose pilot frequency is closest to that of C_3 (1,588 vs. 1,604). On the full training split, 193,874 graphs contain C_3 (5.750%) and 178,620 contain P_6 (5.297%).

I.2 Removal Sets, Matched Removal, and Verification

The ratio of target-withheld to matched-removal hits in Section 5 separates the target's absence from data loss only if no graph retained in a target-withheld set contains its target, and each matched-removal set removes as many target-free graphs, matched on sequence length, while keeping every target-containing graph.

Target withheld and matched removal. The target-withheld set removes every training graph that contains the target, leaving 3,178,084 graphs for C_3 and 3,193,338 for P_6. In the matched-removal control, the sequence-length histograms of the removed target-free graphs and of the target-containing graphs match exactly for C_3. For P_6, the design first required the same exact match, which is infeasible at three lengths (153, 167 and 193) that have fewer target-free than target-containing graphs. Before any P_6 model outcome existed, we replaced only this matching clause: the removed graphs are chosen by an integral one-dimensional minimum-cost flow that minimizes the total absolute length difference subject to a hard caliper of 2 tokens. The solution, certified optimal, has total distance 10: five removed graphs differ in length by 2 tokens and all others match exactly.

Verification. The SHA-256 hashes of the index files that define each training subset match the records of all 15 trained models. An independent re-scan of all 3,371,958 training sequences with a separately written parser and predicates (NetworkX, Hagberg et al., 2008, only for bridge detection) finds 193,874 C_3-containing and 178,620 P_6-containing graphs, with zero graph-level disagreements with the stored flags; no retained graph of either target-withheld set contains its target, and every target-containing graph is present in the corresponding matched-removal set.

What remains in training. Only the exact labeled targets are absent, not three-membered rings or six-carbon single-bond chains in general. The C_3-withheld training set still contains 30,252 graphs with some 3-cycle, which includes a heteroatom (e.g., epoxides and aziridines) or has at least one non-single bond. The P_6-withheld training set still contains 730,906 graphs with a six-carbon single-bond path whose edges are not all bridges.

I.3 Training

Lower target emission after removal could also arise if the target-withheld models were trained differently from the controls or fit the data worse. All conditions share one architecture, recipe and token budget, and same-seed models share their initialization; a post hoc recomputation of the held-out loss (below) checks the fit.

All 15 models (Section 5.1) use the *small* LLaMA-architecture body (Table 4; $N = 113{,}265{,}408$ non-embedding parameters) with the 86-token action vocabulary, trained from random initialization under one recipe (Table 5). The learning-rate schedule is indexed by tokens: linear warmup, then cosine decay that would reach zero at $40N$ tokens. Training stops at $20N$ tokens, where the learning rate is still 0.508 times its peak; all generations use the $20N$ checkpoint. Because the budget is fixed in tokens, the number of passes differs by condition (Table 15).

Held-out loss; no convergence claims. Training stops halfway through the cosine horizon, so we do not describe these models as converged. We computed the held-out NLL of the $20N$ checkpoints (float32, CPU) on the frozen training-time panels of 4,096 target-free validation molecules: all 15 models reach 0.346–0.349 nats per predicted token, and on each panel the conditions differ by at most 0.0021 (Table 16). The target-withheld models thus fit target-free molecules as well as the controls. No hit count or ratio in Section 5 depends on these values; they only rule out poorer fit as an explanation.

Table 15: Token accounting for the 15 target-removal models (five conditions, three training seeds each). Tokens per pass: tokens in one pass over the training set, including the start and end tokens of every sequence. Tokens seen: non-padding tokens consumed when training stops at $20N = 2{,}265{,}308{,}160$ tokens ($N = 113{,}265{,}408$ non-embedding parameters); the value is the same for the three training seeds of a condition, and it exceeds $20N$ by less than one update. Because the budget is fixed in tokens rather than in passes, models trained on the smaller removal sets see each retained graph slightly more often than the full-data models.

Condition	Training graphs	Tokens per pass	Tokens seen	Passes
Full data (shared by C_3 and P_6)	3,371,958	203,647,308	2,265,340,523	11.12
C_3 target withheld	3,178,084	191,530,688	2,265,326,904	11.83
C_3 matched removal	3,178,084	191,530,688	2,265,330,899	11.83
P_6 target withheld	3,193,338	193,072,512	2,265,316,977	11.73
P_6 matched removal	3,193,338	193,072,506	2,265,318,009	11.73

Table 16: Held-out NLL (nats per predicted token) of the 15 target-removal models at the $20N$ checkpoint used for all generations, recomputed after training in float32 on the frozen panels of 4,096 target-free validation molecules. Each model is scored on the panel of its experiment; the three full-data models, shared by both experiments, are scored on both panels.

Panel	Condition	Seed 1	Seed 2	Seed 3
C_3 panel	Target withheld	0.3476	0.3474	0.3473
	Matched removal	0.3485	0.3493	0.3490
	Full data	0.3475	0.3484	0.3488
P_6 panel	Target withheld	0.3457	0.3468	0.3463
	Matched removal	0.3468	0.3475	0.3478
	Full data	0.3465	0.3475	0.3478

I.4 Sampling, Validity, and Novelty

A hit is a strict-valid output that contains the target, and the endpoint counts hits among all raw attempts; the sampler, the validity test and the target predicate (Appendix I.1) therefore determine the hit counts, and the novelty test determines which hits are novel.

Sampler and paired streams. Each model is prompted with <s> and sampled with temperature 1, top-k 50 (Fan et al., 2018) and top-p 0.95 (Holtzman et al., 2020) (not pure ancestral sampling), with a maximum length of 512 tokens including <s>. Each model receives five decoder streams of 32,768 attempts (163,840 raw attempts), with no retries, no early stopping and no filtering; every attempt stays in the denominator. Batch random seeds are derived by hashing the stream and batch indices, not the model, so all 15 models consume the same random numbers at the same draw index (common random numbers). Within one model the draws are independent, so the exact Clopper–Pearson intervals in Tables 18 and 19 are valid Monte Carlo intervals for that fixed model and decoder; they say nothing about variation across training seeds. For each rate ratio, Table 20 combines two-sided 97.5% Clopper–Pearson intervals for the two rates. If these marginal intervals are $[L_1, U_1]$ and $[L_2, U_2]$, the ratio interval is $[L_1/U_2, U_1/L_2]$ (unbounded above if $L_2 = 0$). A union bound gives coverage of at least 95% for each ratio, regardless of the dependence between models induced by shared random numbers; this is not a simultaneous guarantee for all ratios.

Strict validity. Strict validity uses the node/edge-addition parser of Appendix D, with no repair. Between 97.27% and 97.74% of attempts are strict-valid in each of the 15 generation sets; the invalid outputs are mostly disconnected graphs.

Novelty. A strict-valid output is novel if it is not isomorphic, with atom and bond labels, to any of the 3,371,958 graphs of the original, unfiltered training split, including the removed ones; isomorphism is tested as in Appendix D. An independent re-implementation reproduced every novelty flag of the target-withheld hits: 65 of 65 for C_3 and 3,194 of 3,194 for P_6.

Non-novel P_6 hits are withheld molecules. If a strict-valid output contains P_6 and is isomorphic to an original training graph G, then G contains P_6, because containment and the bridge property are preserved under isomorphism; and every P_6-containing training graph was removed, so G is a withheld molecule. A direct lookup confirms this: for the 119, 136 and 119 non-novel hits of the three P_6-withheld models, the isomorphic training graphs are all P_6-containing, and none belongs to the P_6-withheld training set. These are 11.3%, 12.4% and 11.5% of the hits: exact reproductions of whole molecules the models never saw.

I.5 Order of Design Decisions

A target, endpoint or control chosen after seeing outcomes could be tuned to them. We therefore record when each decision was fixed relative to the outcomes it concerns.

1. Before any model was trained: the C_3 target, the three conditions, the removal and matched-removal sets, the model and the training recipe. The primary endpoint then planned was teacher-forced (the log-probability of the triangle-closing actions on fixed prefixes); it was not formally evaluated, and nothing in this paper depends on it.
2. After training had started, but before any C_3 outcome was inspected: the generation endpoint and its decision rules. (a) One strict-valid C_3-containing output from a target-withheld model places that model, with this decoder, outside the bridge-only class of Lemma 5. (b) If no C_3-withheld model produced C_3 while all three P_6-withheld models produced P_6, we would report a finite-budget boundary for the cyclic target, without attributing a mechanism. Rule (a) applies in all three training seeds; rule (b)'s condition is false.
3. After the C_3 models had started training, but before any C_3 or P_6 outcome was inspected: the P_6 experiment and its decision rule (at least one strict-valid P_6-containing output in each P_6-withheld model at 163,840 attempts, novelty not required), which holds in all three seeds.
4. After the C_3 outcome was known: the matched-initialization control, the analysis of Appendix J, the independent re-scan and novelty re-implementation, and (after all P_6 outcomes) the k-gram control (Appendix I.8) and the held-out loss recomputation (Appendix I.3).

I.6 Matched-Initialization Control

To check that C_3 emission is not already present before training, we sampled the three untrained initializations (one per training seed) as a post hoc control, with the same sampler and 163,840 attempts each. They produced no strict-valid output and hence no C_3 (one-sided 95% upper bound on the per-attempt rate: $1 - 0.05^{1/163,840} \approx 1.83 \times 10^{-5}$). The control is uninformative: it cannot separate learning that is specific to the target from learning to emit valid sequences at all.

I.7 Full Results

Table 17 summarizes the hits of every model, and Tables 18–20 give the full results for each training seed, with the Monte Carlo intervals of Appendix I.4.

Matched removal versus full data. Matched removal also raises the target's share of the training set (by the factors below); if emission tracked that share, matched-removal models would emit the target only slightly more often than full-data models. For C_3, in every training seed the matched-removal model emits the target more often than the full-data model (per-seed ratios 1.37, 1.22 and 1.28), whereas the target's share rose only by a factor of 1.061. For P_6 the ratios are 1.05, 1.13 and 1.00, in line with its share increase by a factor of 1.056. We cannot explain the C_3 excess and do not interpret it. It does not change the order of magnitude of the suppression: against full data, C_3 is suppressed by a factor of the same order as against matched removal (Table 20).

Table 17: Generation after target removal (PCQM4Mv2, construction-sequence serialization). Counts are per model, out of 163,840 raw attempts (temperature 1, top-k 50, top-p 0.95); hits and novelty as in Section 5.1; ratio intervals in Table 20. †The same three full-data models and generations serve both targets (the equal seed-2 counts are a verified coincidence).

		Target hits			Novel hits		
Target	Condition	Seed 1	Seed 2	Seed 3	Seed 1	Seed 2	Seed 3
C_3	Target withheld	15	22	28	15	22	28
	Matched removal	11,992	11,866	12,121	9,551	9,607	9,639
	Full data†	8,772	9,707	9,474	6,912	7,798	7,625
	Withheld / matched	*0.13%*	*0.19%*	*0.23%*			
P_6	Target withheld	1,057	1,100	1,037	938	964	918
	Matched removal	9,959	10,981	10,234	7,621	8,270	7,764
	Full data†	9,503	9,707	10,262	7,224	7,480	7,875
	Withheld / matched	*10.6%*	*10.0%*	*10.1%*			

Table 18: C_3 results for all nine models (PCQM4Mv2, construction-sequence serialization, 113.4M-parameter decoder trained for 20 tokens per non-embedding parameter). Each row is one trained model with 163,840 raw attempts (unconditional sampling, temperature 1, top-k 50, top-p 0.95; five decoder streams of 32,768). *Strict-valid*: outputs that parse, without repair, into a connected labeled graph. *Hits*: strict-valid outputs whose graph contains C_3 (three carbons pairwise joined by single bonds). *Hit rate*: hits divided by all raw attempts, invalid outputs included, with an exact (Clopper–Pearson) two-sided 95% interval; the interval describes Monte Carlo variation for one fixed model and decoder, not variation across training seeds. One-sided 95% upper bounds for the target-withheld rates are 0.0141%, 0.0192% and 0.0234%. *Novel*: hits not isomorphic, with atom and bond labels, to any of the 3,371,958 graphs of the original, unfiltered training split; *non-novel* = hits − novel. *RDKit*: hits that pass RDKit sanitization, a diagnostic that never filters outputs.

Condition	Seed	Strict-valid	Hits	Hit rate, % [exact 95% CI]	Novel	Non-novel	RDKit
Target withheld	1	160,132	15	0.00916 [0.00512, 0.01510]	15	0	13
	2	160,126	22	0.01343 [0.00842, 0.02033]	22	0	18
	3	159,778	28	0.01709 [0.01136, 0.02470]	28	0	24
Matched removal	1	160,036	11,992	7.319 [7.194, 7.447]	9,551	2,441	11,171
	2	159,719	11,866	7.242 [7.117, 7.369]	9,607	2,259	11,047
	3	160,038	12,121	7.398 [7.272, 7.526]	9,639	2,482	11,250
Full data	1	159,361	8,772	5.354 [5.245, 5.464]	6,912	1,860	8,166
	2	159,922	9,707	5.925 [5.811, 6.040]	7,798	1,909	9,017
	3	159,897	9,474	5.782 [5.670, 5.897]	7,625	1,849	8,696

I.8 Exact-Count k-gram Control

This control tests the local alternative of Section 5.3. It was designed after all Transformer outcomes were known; its models, decoders, endpoint and reporting rules were fixed in a dated design record before any k-gram was estimated, and the single run is reported in full. An addendum, written after the C_3 k-gram results and before any P_6 k-gram was sampled, applies the same control to P_6.

Models and sampling. On each training set of the C_3 and P_6 experiments (target withheld, matched removal, full data) we estimated exact-count k-gram models over the 86 construction-sequence tokens, $k \in \{3, 5, 8\}$, defined like the DFS-code n-grams of Appendix F: maximum-likelihood counts, no smoothing and no back-off, with contexts padded by a begin symbol. Each model was sampled like the Transformers (Appendix I.4); the 5-gram was also sampled with pure ancestral sampling. Stream seeds are shared by the three training sets. Outputs are scored with the same strict parser, target predicates and novelty test as the Transformer outputs; novelty is decided against the 193,874 C_3-containing or 178,620 P_6-containing training graphs, which suffices because labeled isomorphism preserves containment. Every k-token window of every hit of a target-withheld

Table 19: P_6 results; columns and sampling as in Table 18. A hit is a strict-valid output whose graph contains P_6: six carbons joined by five single bonds in a path whose edges are all bridges of the output graph. For the target-withheld models, every non-novel hit is isomorphic to a training molecule that had been withheld (Appendix I.4). Distinct labeled graphs among the target-withheld hits: 937, 962 and 917 novel; 118, 135 and 115 non-novel. †The same three full-data models and generations as in Table 18, scored for P_6 instead of C_3; strict-valid counts are therefore identical. In seed 2 the C_3 and P_6 hit counts coincide (9,707); they were computed from separate per-output flags.

Condition	Seed	Strict-valid	Hits	Hit rate, % [exact 95% CI]	Novel	Non-novel	RDKit
Target withheld	1	159,726	1,057	0.645 [0.607, 0.685]	938	119	994
	2	159,914	1,100	0.671 [0.632, 0.712]	964	136	1,041
	3	159,788	1,037	0.633 [0.595, 0.673]	918	119	973
Matched removal	1	159,577	9,959	6.078 [5.963, 6.195]	7,621	2,338	9,479
	2	159,882	10,981	6.702 [6.582, 6.824]	8,270	2,711	10,385
	3	159,943	10,234	6.246 [6.130, 6.365]	7,764	2,470	9,726
Full data†	1	159,361	9,503	5.800 [5.687, 5.914]	7,224	2,279	9,064
	2	159,922	9,707	5.925 [5.811, 6.040]	7,480	2,227	9,197
	3	159,897	10,262	6.263 [6.147, 6.382]	7,875	2,387	9,683

Table 20: Per-seed ratio of the hit count of the target-withheld model to that of the matched-removal or full-data model with the same training seed (all denominators are 163,840 raw attempts, so count ratios equal rate ratios). Each interval combines two-sided 97.5% Clopper–Pearson intervals for the two rates; by the union bound it has coverage of at least 95% whatever the dependence between models, per ratio and not simultaneously for all 12 (Appendix I.4). Like the per-model intervals, these reflect Monte Carlo variation for fixed models, not variation across training seeds; we do not pool seeds.

Target	Seed	Withheld / matched removal [95% CI]	Withheld / full data [95% CI]
C_3	1	0.00125 [0.00063, 0.00223]	0.00171 [0.00086, 0.00307]
	2	0.00185 [0.00106, 0.00301]	0.00227 [0.00129, 0.00369]
	3	0.00231 [0.00142, 0.00357]	0.00296 [0.00181, 0.00458]
P_6	1	0.106 [0.0968, 0.116]	0.111 [0.101, 0.122]
	2	0.100 [0.0916, 0.109]	0.113 [0.104, 0.124]
	3	0.101 [0.0924, 0.111]	0.101 [0.0922, 0.111]

model occurs in its target-withheld training set, and independent implementations of both predicates confirm every hit.

Results. Every k-gram emits C_3 after removal, and removal suppresses it far less than in the Transformers: the target-withheld models keep 21–116% of their matched-removal hit count, against 0.13–0.23% for the Transformers (Table 21). Longer contexts suppress more (3-gram 116%, 5-gram 44%, 8-gram 21%), and pure sampling does not change the picture (5-gram 60%). The k-grams are poor generators overall (0.7–25.5% of attempts are strict-valid, against 97.3–97.7%), so we compare training sets within each generator, not raw rates across generators. Their hits also differ in kind: 9–50% pass RDKit sanitization, against 55 of the Transformers' 65. Three of the 5-gram's 58 hits are labeled-isomorphic to withheld training molecules. For P_6 the 5- and 8-grams keep 77% and 86% (the 3-gram emits only 2 and 5 hits), against 10.0–10.6% for the Transformers. Local token statistics thus register the removal of either target far more weakly than the Transformers, which suppress P_6 about tenfold and C_3 several hundredfold.

I.9 Proof of Lemma 5

We use the notation of Lemma 5. Let $G = (V, E)$ be an output, obtained from the disjoint union of fragments $F_1, \ldots, F_m$ by adding a set J of labeled edges. Then $V = V(F_1) \sqcup \cdots \sqcup V(F_m)$ and $E = E(F_1) \sqcup \cdots \sqcup E(F_m) \sqcup J$, every edge of $E(F_i)$ joins two vertices of $V(F_i)$, and by assumption

Table 21: Target emission after removal: the Transformers of Table 17 and the pre-specified exact-count k-gram control (Appendix I.8). Every generator draws 163,840 raw attempts from each training set with no filtering, with the decoder of Section 5 (temperature 1, top-k 50, top-p 0.95) except the pure-sampling 5-gram. Strict-valid and sanitizable: target-withheld model; hits: strict-valid outputs that contain the target (novel in parentheses); ratio intervals as in Table 20. The k-grams are deterministic count models with shared random streams across training sets; they have no training seeds.

	Strict-valid	Target hits			Withheld / matched	Sanitizable
Generator	(withheld)	Withheld (novel)	Matched	Full	% [95% CI]	(withheld)
C_3						
Transformer, seed 1	160,132	15 (15)	11,992	8,772	0.13 [0.06, 0.22]	13
Transformer, seed 2	160,126	22 (22)	11,866	9,707	0.19 [0.11, 0.30]	18
Transformer, seed 3	159,778	28 (28)	12,121	9,474	0.23 [0.14, 0.36]	24
3-gram	1,081	22 (11)	19	16	116 [41, 331]	11
5-gram	5,601	58 (55)	132	122	44 [26, 72]	12
8-gram	41,787	516 (516)	2,507	2,414	21 [18, 24]	47
5-gram, pure sampling	4,272	59 (56)	99	103	60 [35, 101]	14
P_6						
Transformer, seed 1	159,726	1,057 (938)	9,959	9,503	10.6 [9.7, 11.6]	994
Transformer, seed 2	159,914	1,100 (964)	10,981	9,707	10.0 [9.2, 10.9]	1,041
Transformer, seed 3	159,788	1,037 (918)	10,234	10,262	10.1 [9.2, 11.1]	973
3-gram	1,114	2 (2)	5	4	40 [1, 600]	0
5-gram	5,086	356 (325)	463	454	77 [61, 96]	133
8-gram	40,857	1,842 (1,803)	2,136	2,168	86 [78, 95]	615
5-gram, pure sampling	3,858	264 (232)	323	315	82 [63, 107]	95

every $e \in J$ is a bridge of G. For each i there is $D_i \in \mathcal{D}$ and a label-preserving monomorphism $g_i : F_i \to D_i$.

We use the standard fact that an edge $e = \{u, v\}$ is a bridge of G if and only if it lies on no cycle of G: if e lies on a cycle, the rest of the cycle is a u–v path avoiding e, so deleting e disconnects nothing; if e is not a bridge, a u–v path in $G - e$ forms a cycle with e.

Proof of Lemma 5. Suppose, for contradiction, that some output G contains C_3, and let f be a label-preserving monomorphism from C_3 into G. Its image is a triangle T with carbon vertices x, y, z and single-bond edges $\{x, y\}, \{y, z\}, \{z, x\}$. (i) Each edge of T lies on the cycle T, so by the fact above none of them is a bridge of G; hence no edge of T is in J, and each lies in some $E(F_i)$. (ii) An edge in $E(F_i)$ has both endpoints in $V(F_i)$, and the sets $V(F_i)$ are disjoint, so the edges $\{x, y\}$ and $\{y, z\}$ place x, y, z in $V(F_i)$ for a single i. An edge of $E(F_j)$ with $j \neq i$ cannot join vertices of $V(F_i)$, so all three edges of T lie in $E(F_i)$, and f is a label-preserving monomorphism from C_3 into F_i. (iii) Composing with g_i gives a label-preserving monomorphism $g_i \circ f$ from C_3 into $D_i \in \mathcal{D}$, so D_i contains C_3. This contradicts the assumption that no graph in $\mathcal{D}$ contains C_3. □

Remarks. (1) The proof uses only that the pattern is connected and that each of its edges lies on a cycle of the pattern, so the conclusion holds for any such pattern (e.g., any ring), but not for a pattern with a bridge, whose bridge edges may be added edges; for P_6 it fails (see below). (2) The bridge-only baseline of Section 4.3 is in the class: each output joins two fragments of training graphs by one new bond, which is a bridge of the output (Appendix F).

P_6 lies inside the class: three witnesses. Before any P_6 outcome, we validated three explicit constructions of P_6 by the implemented crossover from retained, P_6-free training graphs. In each, two distinct retained training graphs are each cut at a single-bond bridge; the kept side of each contains an all-carbon, single-bond path of bridges of its source graph that ends at the cut atom. Joining the two cut atoms by a new single bond yields a connected graph in which the new bond is a bridge and a six-carbon path through it satisfies the P_6 predicate (checked with the same predicate

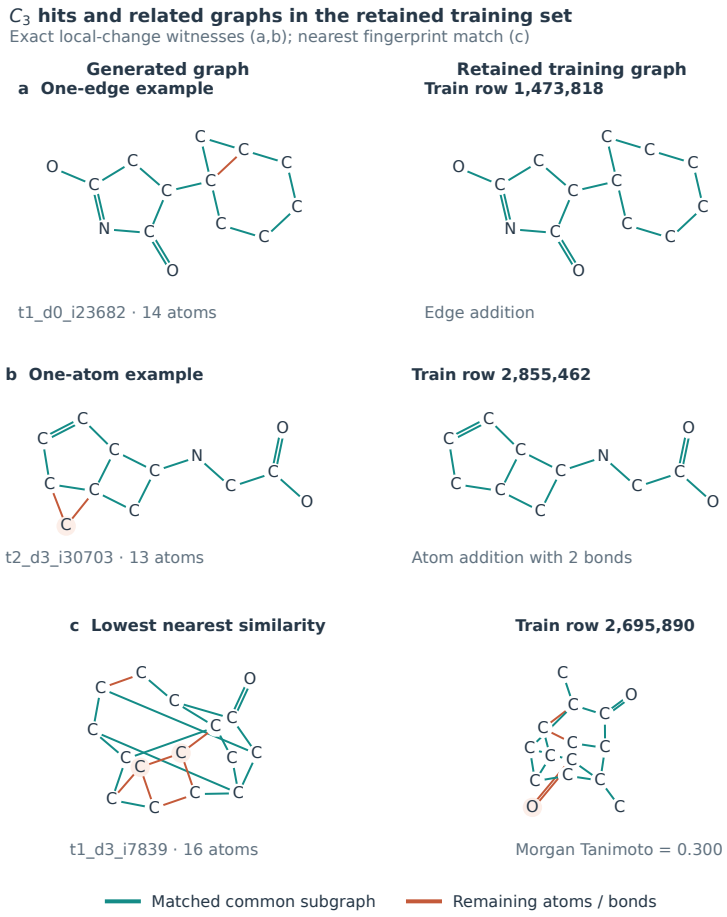


Figure 3: Three C_3 hits of target-withheld models (left) next to graphs from the retained, C_3-free training set (right). Teal: matched atoms and bonds; orange: the rest; line multiplicity: bond order. (a) The first hit, in a fixed order of output identifiers, with an edge-change witness: the retained graph plus one C–C single bond, which closes the triangle. (b) The first hit with only an atom-change witness: the retained graph plus one carbon bonded by single bonds to two adjacent carbons. In (a) and (b) the mapping is the exact witness. (c) The hit least similar to the retained training set: among the 55 RDKit-sanitizable hits, the one whose most similar retained graph by Morgan fingerprint (radius 2, 2,048 bits) has the lowest Tanimoto similarity (0.30); the coloring is an illustrative RDKit common-subgraph mapping, not an exact witness. The selection is deterministic, not a random sample; neither the drawings nor the fingerprint score measure graph edit distance.

code). The output support of the bridge-only class thus contains P_6-containing graphs: P_6 emission by the P_6-withheld models is compatible with that class, whereas C_3 emission is not.

J THE 65 TRIANGLE-CONTAINING OUTPUTS

Section 5.3 asks whether a hit could be a small edit of a retained training graph. This post hoc analysis uses only the saved C_3 hits of the three target-withheld models; nothing was retrained or regenerated. The hits are 65 distinct labeled graphs with 9–30 atoms (median 17), none isomorphic, with labels, to any original training graph (Appendix I.4).

One local change. We compared each hit with every retained training graph, asking whether one local change of the kinds defined in Table 22 turns one into the other. Of the 65 hits, 10 are one such change away from a retained training graph (5 by an edge change, 5 only by an atom change), and

the other 55 are not reachable by any enumerated single change. Since the retained graphs contain no C_3, every witness must create the triangle. All five edge changes add the C–C single bond that closes it (Figure 3a), and in every recorded atom-change witness the added atom is a carbon of the triangle (Figure 3b). These 10 hits are therefore compatible with a local-edit explanation, which we do not exclude. For the other 55, the result excludes only the enumerated single changes; it does not establish a large graph edit distance from the training set, and it does not identify how the outputs were generated.

Table 22: Local-change comparison of the 65 C_3 hits of the target-withheld models with all 3,178,084 retained training graphs (post hoc; categories are mutually exclusive). *One edge or label change*: some retained graph differs from the hit by inserting, deleting or relabeling one edge, or relabeling one atom. *One atom change only*: no such witness, but some retained graph differs by inserting or deleting one atom together with all of its incident edges (not a unit graph edit). *Neither*: no enumerated single change reaches any retained graph. Comparing with the full original training set (3,371,958 graphs) gives the same counts.

Training seed	C_3 hits	One edge or label change	One atom change only	Neither
1	15	2	0	13
2	22	1	1	20
3	28	2	4	22
Total	65	5	5	55

Chemical sanitization. Strict validity is structural; RDKit sanitization is a chemical diagnostic, and Section 5.3 uses its pass counts to contrast Transformer and k-gram hits. 55 of the 65 hits pass RDKit sanitization (RDKit, 2026); the other 10 are strict-valid graphs that fail it but remain in all counts. The 55 sanitizable hits and the 55 hits without a single-change witness are different sets that happen to have the same size.

K Likelihood and Support: Details

This appendix details Section 6. Unless stated otherwise, analyses use PCQM4Mv2 DFS codes (Section 3.2), the fixed panel of 253 patterns (Section 3.3) counted exactly by VF2, the supports of 200,000 training graphs as reference ("train-200k"; 14.14 nodes on average), and pure ancestral sampling with three generation seeds per checkpoint, averaged. The error $\overline{|\Delta s|}$ and its valid-only, novel-only and total-map conventions are those of Section 3.3. Held-out NLL is the mean negative log-likelihood in nats per target (tuple or end token) on 20,000 validation molecules (344,043 targets) stratified into 10 strata of edge-tuple count, after removing strict-parser failures, strings that occur verbatim in training, and duplicates; this length-stratified, training-disjoint panel gives a common scale for models, not an estimate of the population cross-entropy.

Checkpoints and reference levels. The checkpoints span model size and training length, so that NLL varies with both. Eight Transformer runs share one recipe (Tables 4 and 5; peak learning rate 10^{-5}): five small runs (132.2M parameters; four distinct training seeds and four distinct data orders) and one run each of tiny (10.5M), medium (99.1M) and large (451.0M; 12 epochs, not converged). Each run was sampled at six checkpoints log-spaced in tokens seen (48 in total; 16,384 raw attempts per generation seed for small, 8,192 otherwise). At $n = 16{,}384$ valid graphs, a held-out resample scores $\overline{|\Delta s|} = 0.0078$, about 90% of it the fixed support shift between the training and held-out splits (0.0070). It is a reference, not a noise floor: a model below it is not thereby closer to the data distribution than real molecules are. As $\overline{|\Delta s|}$ shrinks with n, the *n-matched reference* is the held-out resample at a model's own number of valid graphs.

K.1 Association Between Held-Out NLL and Support Error

We test whether, within one recipe, lower held-out NLL goes with lower support error (Section 6). Because NLL, tokens seen and model size move together in training, a raw correlation between NLL and an error could arise merely from training length and size, so we correlate residuals: we regress

the unranked NLL and each unranked endpoint (the mass-weighted total-map error, the valid-only and novel-only errors, and the validity rate) separately on $\log T$, $(\log T)^2$ (T = tokens seen), model-size indicators and an intercept, then compute Spearman correlation between the OLS residuals. This differs from rank-based partial correlation, which ranks variables before residualizing. Percentile 95% intervals come from a bootstrap over the eight training runs (2,000 draws); runs sharing a seed or data order are separate clusters, so that dependence is ignored. A positive correlation describes residual association after these adjustments, not a causal effect.

Total map. Across the 48 checkpoints, the mass-weighted total-map error (patterns weighted by their reference support) has Pearson correlation 0.99995 with the invalid fraction and raw Spearman correlation 0.994 with NLL, but neither it (residual $\rho_S = -0.325$, CI $[-0.80,\ 0.48]$) nor the validity rate ($+0.32$, $[-0.48,\ 0.81]$) shows a detectable residual association with NLL (with eight run clusters, the intervals are wide): beyond training length and size, NLL carries no detectable signal about the invalid mass that this error tracks. To check how much of this error invalid mass alone explains, we compare it with a *q-matched reference* (a held-out resample padded with empty graphs to the model's invalid mass q); across the 48 checkpoints the largest absolute deviation is 0.0036. This is consistent with a large validity contribution in Lemma 3; matching this absolute error alone does not establish that valid-only errors vanish.

Valid-only. Among the 47 checkpoints with valid outputs, the valid-only error has raw Spearman correlation 0.985 with NLL and residual correlation 0.919, CI $[0.736,\ 0.984]$ (Figure 4a). To check that exact training-string copies do not drive this, we repeat the analysis on the novel-only error (46 checkpoints): residual correlation 0.861, $[0.633,\ 0.952]$. NLL and generated size (mean node count of valid outputs) are collinear here ($\rho_S(\text{mean nodes}, \text{error}) = -0.905$, $\rho_S(\text{mean nodes}, \text{NLL}) = -0.919$), so this family alone cannot tell which of the two indexes the error. To see whether the relation depends on the decoder, we also sampled under top-k 50 and top-p 0.95 at temperature 1 (a separate, unpooled sensitivity check; Fan et al., 2018; Holtzman et al., 2020): the error is flat at 0.020–0.028 across all 30 checkpoints with NLL ≤ 1.442, so, in this NLL range, the rise of valid-only error with NLL appears under pure ancestral sampling but not under this truncated decoder, consistent with Section 4.2.

K.2 Other Generators at Matched NLL

To examine whether this association extends beyond the main recipe, we compare three other generators with the Transformer relation at nearby NLL (Table 23): Witten–Bell n-grams, two LSTMs, and the same Transformer at a 30× larger learning rate (not in Figure 4).

Witten–Bell n-gram models. We ask whether a non-neural generator at matched NLL has comparable support error. Unlike the unsmoothed, rejection-sampled exact-count n-grams of Section 4.3, which assign zero probability to unseen continuations, smoothed models have a finite held-out NLL. The order-k model conditions on the k preceding tuple tokens (the exact-count k-grams of Table 1 condition on $k-1$) and predicts the next tuple or the end token with interpolated Witten–Bell smoothing (Witten & Bell, 1991; Chen & Goodman, 1999) down to an add-0.5-smoothed unigram. All orders are trained on the 3,371,958 training strings and sampled exactly (no truncation, no rejection). Section 6 uses order 5. Orders 1, 2 and 4 (not in Table 23) reach NLL 2.083, 1.738 and 1.504 with valid-only error 0.0829, 0.0462 and 0.0264. The order-5 model (valid-only error 0.0162) is comparable to the four Transformer checkpoints within ± 0.05 nats of its NLL (0.0115–0.0188); this is a descriptive comparison, not an equivalence test. This raw comparison, at each model's own n, favors the order-5 model (more valid outputs). Within node-count bins (valid graphs resampled to a common $n = 300$, reference restricted to the bin), its residual from a local Transformer line is ≤ 0 in all six bins, which, given the sampling noise at this n, we read only as no evidence that it lies above the Transformer relation in any size class. The order-8 model's low error is a copying effect: on novel-only outputs its residual is $+0.016$, above the relation. At matched NLL the n-gram models are valid more often (0.43–0.79) than the Transformers (0.16–0.46), so their total-map errors differ even where valid-only errors agree. As the Transformer relation is recipe-specific (learning-rate probe below), we claim no relation shared with the n-gram models.

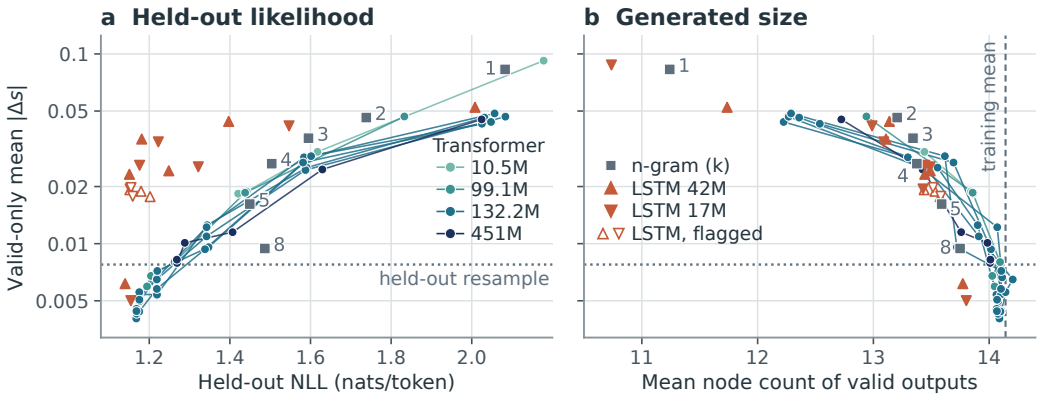


Figure 4: Held-out likelihood does not fix support error across model classes. Valid-only $\overline{|\Delta s|}$ (log scale) against **(a)** held-out NLL per DFS token and **(b)** the mean node count of valid outputs (dashed: training mean, 14.14). Transformers of one recipe (peak learning rate 10^{-5}; 47 checkpoints of eight runs; thin lines join one run), Witten–Bell n-grams (labeled by order k) and two LSTMs (one training seed each; open markers: exact-string novelty below 0.5 or NLL higher than at the previous checkpoint). Dotted: held-out resample reference (0.0078 at 16,384 graphs). Five early checkpoints with NLL above 2.2 lie outside (a), five with mean node count below 10.5 outside (b).

Table 23: Representative checkpoints and Witten–Bell orders 3, 5 and 8. NLL: nats per target; validity: fraction of raw attempts; novelty: fraction of valid outputs that are not exact training strings; mean nodes: over valid outputs. Means over three generation seeds of 8,192 raw attempts (99M Transformer, n-grams) or 16,384 (others); ranges over the five small runs. The local main-recipe relation predicts 0.0084 at NLL 1.260 and 0.0051 at 1.188.

Model and checkpoint	NLL	Validity	Novelty	Mean nodes	Valid-only $\overline{\lvert\Delta s\rvert}$
Transformer, main recipe (peak learning rate 10^{-5})					
132M, final (5 runs)	1.168–1.169	0.858–0.861	0.754–0.761	14.06–14.11	0.0040–0.0045
132M, step 11,508 (5 runs)	1.175–1.176	0.849–0.855	0.766–0.777	14.07–14.14	0.0044–0.0056
99M, step 3,288	1.438	0.346	0.870	13.86	0.0186
Witten–Bell n-gram, order k (number of preceding tuples)					
order 3	1.595	0.428	0.939	13.34	0.0360
order 5	1.449	0.621	0.840	13.59	0.0162
order 8	1.487	0.790	0.585	13.75	0.0094
LSTM (one training seed each)					
42M, epoch 2.49	1.181	0.921	0.770	13.11	0.0356
17M, epoch 4.98	1.176	0.926	0.764	13.44	0.0259
Transformer 132M, peak learning rate 3×10^{-4} (one training seed)					
epoch 0.5	1.260	0.809	0.787	13.12	0.0252
epoch 1.0	1.188	0.858	0.763	13.66	0.0110

LSTM models. We ask whether the LSTMs lie on the Transformer relation at matched NLL and, if not, whether NLL or generated size indexes the gap. Two two-layer LSTMs (Hochreiter & Schmidhuber, 1997) of hidden width 1,024 ("LSTM 42M") and 512 ("LSTM 17M") use the Transformers' training split and tuple tokens. Each was trained once (AdamW, learning rate 10^{-3}, cosine decay, no dropout; Loshchilov & Hutter, 2019) with ten checkpoints between epochs 0.1 and 50; flagging those with novelty below 0.5 (copying) or NLL higher than at the previous checkpoint (overfitting) leaves 14 unflagged. We compare each checkpoint's excess of valid-only error over the n-matched reference with that of the four Transformer checkpoints nearest in NLL. Twelve of the 14 unflagged are flagged as above this reference by exploratory Welch comparisons (p from 1.4×10^{-7} to 2.9×10^{-3}), including all seven inside the Transformer NLL range (the other five are extrapolations); the two exceptions (epoch 9.96) have NLL below every Transformer checkpoint. These nominal tests compare three generation seeds of one LSTM checkpoint with four Transformer checkpoint means (different

units; the checkpoints need not be independent), so they are exploratory diagnostics, not calibrated tests of architecture-level differences. The LSTMs generate smaller graphs (13.089–13.803 mean nodes at NLL $\leq$ 1.33, against 13.985–14.204 for Transformers at NLL $\leq$ 1.30). Predicting each of the 16 LSTM checkpoints from epoch 0.5 on by the log-space average of its four nearest Transformer checkpoints (NLL < 2) gives a mean absolute log-ratio of 1.190 with nearness in NLL but 0.323 with nearness in mean node count, close to the Transformers' own leave-one-out scatter under node count (0.261). Each LSTM is one training run with its own recipe, and 7 of these 16 checkpoints have NLL below the lowest Transformer NLL (not truly matched). NLL thus does not fix the generated-size distribution across model classes; this is not a comparison of the two architectures.

Learning-rate probe. To check whether the NLL–valid-only error relation carries over to another recipe with architecture and data held fixed (which bounds how far Section 6 can state it), we retrained the 132.2M model on the same data (one training seed) with peak learning rate 3×10^{-4}, batch size 512 and 6 epochs. Its two checkpoints inside the main recipe's NLL range lie above that recipe's relation (Table 23) but are mid-schedule, near peak learning rate, so learning rate and schedule position are confounded. Hence we do not treat the NLL–valid-only error relation as holding beyond the main recipe, or as a property of the architecture at a given NLL.

K.3 Why the Pinsker Bound Is Too Loose to Explain the Observed Agreement

We give the calculation behind Consequence (c) of Section 3.1, so that the observed support agreement is not read as a consequence of a small sequence-level divergence $\mathrm{KL}(P_X \| Q_X)$.

Lemmas 2 and 6 and Pinsker's inequality (Cover & Thomas, 2006; Tsybakov, 2009) give, for every pattern H,

$$|s_H(Q) - s_H(P)| \leq \mathrm{TV}(P, Q) \leq \mathrm{TV}(P_X, Q_X) \leq \sqrt{\mathrm{KL}(P_X \,\|\, Q_X)/2}.$$

This bounds population total-map supports; empirical support differences also contain sampling error. KL here is in nats per sequence, not the held-out per-token NLL above. In the ideal setup, $P(\perp) = 0$, so data processing by the parse-validity indicator gives $\mathrm{KL}(P_X \| Q_X) \geq -\log(1 - q)$, where $q = Q(\perp)$.

For the raw validation data, write $\widetilde{P}_X$ for its empirical string distribution, which has a small parser-invalid fraction $p_\perp$, unlike the ideal P_X. The same argument gives $\mathrm{KL}(\widetilde{P}_X \| Q_X) \geq d(p_\perp \| q)$, where d is binary KL; the support bound then concerns $\psi_\# \widetilde{P}_X$. For the headline model, the estimated invalid fraction is $\hat{q} = 0.14155$ (five generation seeds of 4,096 raw attempts). Using this estimate and $p_\perp \leq 3.15 \times 10^{-4}$ gives a plug-in diagnostic of $d \approx 0.1503$ and $\sqrt{d/2} \approx 0.274$, roughly twice the scale of the observed total-map deviation against train-200k (0.138, five-seed mean of the panel maximum). Since $d(p\|q)$ decreases in $p < q$, using this upper bound on $p_\perp$ lowers the diagnostic. The value 0.274 estimates a lower limit on the Pinsker right-hand side, not a lower bound on support error or a certified population bound: uncertainty in $\hat{q}$ has not been propagated. This comparison indicates that this route is too loose to explain the observed agreement; it uses no NLL estimate.