Coarse-to-Fine Autoregression over Hierarchical Discrete Codes for Molecular Graph Generation

Haozhuo Zheng **Cheng Wang** **Pengyu Chen** **Yajun Tian** **Yang Liu***
Harbin Institute of Technology
{24b903065, 21s003047, tianyajun}@stu.hit.edu.cn
pengyuchen603@gmail.com, liuyang@hit.edu.cn

Abstract

Molecular graph generation typically involves a trade-off between two paradigms: diffusion models capture global structure well but require hundreds of denoising steps, while autoregressive (AR) models sample in a single pass yet generate atoms in a flat canonical order with no semantic hierarchy, leaving long-range topology to chance. We introduce the **Hierarchical Graph VQ-Transformer (H-GVT)**, which removes this dichotomy by performing coarse-to-fine AR generation over hierarchical discrete codes. A **Spectrally-Regularized Multi-Scale VQ-VAE** first compresses each molecule into a sequence of discrete tokens at multiple scales, where coarse tokens provide regularized low-resolution structural context and fine tokens encode atom-level details. The compression combines Adaptive Min-Cut Pooling with a novel **Spectral Topology Consistency Loss** that aligns low-frequency normalized Laplacian spectra across scales, adding only 2.8% training overhead. A standard decoder-only Transformer with level embeddings then generates these tokens from coarse to fine, allowing fine-grained atom tokens to condition on coarser structural context. This requires no specialized tree, motif, or scale-causal masking. On QM9 and ZINC250k, H-GVT achieves the best NSPDK among compared baselines (2×10^{-4} on QM9 and 1×10^{-4} on ZINC250k), while sampling 10K molecules $68\times$ faster than DiGress. On MOSES, H-GVT achieves $4.8\times$ better FCD than a Novelty-thresholded GVT baseline under the same thresholded protocol (0.19 vs. 0.92; 86.4% vs. 80.5% Novelty). Crucially, ablations reveal that the coarse-to-fine ordering itself, not merely multi-scale tokenization, is what enables global structural planning: randomizing the order while keeping the same tokens, codebook, and level embeddings degrades NSPDK by $29\times$. Code, configurations, and checkpoints are available at `https://github.com/zzccppp/H-GVT`.

1 Introduction

The de novo generation of molecular graphs with desirable properties is a central task in drug discovery and materials science. Two dominant paradigms have emerged: diffusion and score-based models [Jo et al., 2022, Vignac et al., 2022, Yan et al., 2023, Boget et al., 2025, Jung et al., 2026], which generate graphs by reversing learned noising processes and excel at capturing global structural distributions, and autoregressive (AR) or sequential latent models [You et al., 2018, Liao et al., 2019, Shi et al., 2020, Luo et al., 2021, Boget et al., 2023, Zheng et al., 2026], which construct graphs or graph latents sequentially and offer efficient single-pass sampling. Neither paradigm is fully satisfactory: diffusion models require hundreds of iterative steps (e.g., DiGress takes $\sim$1,000s for 10K molecules), while flat AR models, despite their speed, generate atoms in a fixed canonical traversal order (BFS or DFS) that carries no semantic hierarchy.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

The problem with flat autoregressive generation. In existing AR methods, the first token in a sequence is simply the first atom encountered during graph traversal, not a coarser abstraction of the molecule. This forces the model to commit to local atomic decisions before establishing global topology: it must decide specific atom types and bond orders without a "blueprint" of the overall molecular skeleton. Empirically, this leads to poor global structural coherence. For example, the Graph VQ-Transformer (GVT) [Zheng et al., 2026], despite achieving near-perfect reconstruction, has an NSPDK score of 6.89×10^{-2} on ZINC250k, more than an order of magnitude worse than the best diffusion models, reflecting its inability to plan macro-structures such as ring systems and long-range connectivity. Moreover, its reported strong distribution-matching scores come at the cost of low generation diversity (Novelty as low as 61%), suggesting that flat AR models compensate for structural weakness through memorization rather than true distributional learning.

Key insight. We observe that if molecular graphs are first compressed into discrete codes at multiple scales, where coarse codes provide lower-resolution structural context and fine codes capture individual atomic details, then a standard autoregressive Transformer can implement coarse-to-fine generation simply by ordering these codes from coarse to fine. In our design, every scale is extracted directly from the original graph, and the coarse-to-fine dependency is imposed only at generation time: each fine-scale token is predicted conditional on all coarser predecessors, so global structure is committed before local atomic details. Hierarchy itself is not new in molecular generation: chemically motivated decompositions have been used by JT-VAE [Jin et al., 2018], HierVAE [Jin et al., 2020], and motif-based methods such as MoLeR [Maziarz et al., 2021]. Our contribution lies in replacing these predefined chemical hierarchies with a fully learned multi-scale graph pooling, and in showing that the resulting coarse-to-fine ordering is enough for a standard decoder-only Transformer (with no specialized tree or motif decoder) to plan global topology before local chemistry. Unlike flat AR ordering where "earlier" tokens are merely earlier in a graph traversal, our coarse-to-fine ordering is structurally organized.

Challenges. Realizing this idea for molecular graphs (in contrast to images, where multi-scale generation is well-studied [Tian et al., 2024]) presents three specific challenges. First, molecular graphs vary widely in size and topology, which prevents the definition of uniform fixed-size scales. Second, downsampling a graph requires merging discrete nodes while preserving chemical validity, and naive clustering destroys semantic integrity. Third, without explicit supervision at intermediate scales, pooling modules tend to collapse into trivial clusterings that lose global structural information.

Our approach. We introduce the Hierarchical Graph VQ-Transformer (H-GVT) to address these challenges. H-GVT is a two-stage framework. The first stage is a Spectrally-Regularized Multi-Scale VQ-VAE that compresses molecular graphs into hierarchical discrete codes using Adaptive Min-Cut Pooling (addressing Challenges 1–2) regularized by a novel Spectral Topology Consistency Loss (addressing Challenge 3). The second stage is a standard decoder-only Transformer that generates these codes in coarse-to-fine order via level embeddings, realizing the coarse-to-fine paradigm without any architectural modifications to the Transformer itself.

Contributions. We formulate coarse-to-fine molecular graph generation as autoregressive decoding over hierarchical discrete codes, reaching a new Pareto-optimal point on the speed–quality frontier: state-of-the-art NSPDK (2×10^{-4} on QM9, 1×10^{-4} on ZINC250k) with 68$\times$ faster inference than the leading diffusion model DiGress. To support this formulation, we design a Spectral Topology Consistency Loss that aligns the low-frequency normalized Laplacian spectra across scales, providing a principled and lightweight regularizer (2.8% training overhead) that yields a 3$\times$ reduction in NSPDK error on ZINC250k. On MOSES, the coarse-to-fine paradigm shifts the Pareto frontier of fidelity versus diversity: at matched Novelty ($\geq$80%), H-GVT achieves 4.8$\times$ better FCD than a Novelty-matched GVT baseline (0.19 vs. 0.92) and operates at higher Novelty (86.4% vs. 80.5%). Finally, systematic ablations show that the coarse-to-fine token ordering itself is essential: randomizing the order degrades NSPDK by 29$\times$, indicating that the slot-indexed hierarchical ordering carries useful structural information beyond flat AR ordering.

2 Related Work

2.1 Molecular Graph Generation

Molecular graph generation follows two dominant paradigms. Earlier graph-based molecule generators span VAE, compressed-representation, GAN, conditional, and flow formulations [Simonovsky

and Komodakis, 2018, Samanta et al., 2020, Kwon et al., 2019, 2020, De Cao and Kipf, 2018, Li et al., 2018, Shi et al., 2020, Luo et al., 2021], which established molecular graph generation as a direct alternative to string-based molecule modeling. *Flat AR models* [You et al., 2018, Liao et al., 2019, Shi et al., 2020, Luo et al., 2021, Mercado et al., 2021] build graphs step-by-step under a fixed canonical traversal; recent atom-level hybrids add discrete latents—DGAE [Boget et al., 2023] via vector quantization (with notable information loss) and GVT [Zheng et al., 2026] via RCM ordering with rotary embeddings (the strongest AR baseline)—while SMILES decoders [Segler et al., 2018, Bagal et al., 2021] factorize over a flat string vocabulary. Beyond molecular generation, vector-quantized graph representation learning has also been explored for bridging GNN and MLP representation spaces [Yang et al., 2023]; H-GVT uses the codebook instead as a generative, multi-scale molecular latent. Although tokenization or canonical orderings provide implicit cues, none of these methods explicitly factorize generation across structural scales. *Diffusion models* [Jo et al., 2022, Vignac et al., 2022, Yan et al., 2023, Boget et al., 2025] reverse a noise process and capture global structure well, with PARD [Zhao et al., 2024] mixing block-wise AR and diffusion and MolHIT [Jung et al., 2026] introducing hierarchical discrete diffusion; their iterative sampling, however, is orders of magnitude slower than AR. H-GVT targets the gap between these two regimes.

Hierarchy-aware molecular generation. Chemically motivated hierarchies have long been used: JT-VAE [Jin et al., 2018] decodes a junction tree of rings/bonds before assembly, HierVAE [Jin et al., 2020] attaches motifs from a mined vocabulary, and MoLeR [Maziarz et al., 2021] extends scaffolds with predefined motifs. These two-level (motif-then-atom) schemes improve validity over flat decoding but rely on a frozen, dataset-specific motif inventory. More recent hierarchical denoising-based methods, including HDFM [Boget et al., 2026] (discrete flow matching) and GraphBFN [Xiong et al., 2025] (Bayesian flow networks), also generate graphs coarse-to-fine but inherit the iterative-sampling cost of their underlying denoising frameworks. H-GVT differs in two respects: (i) the hierarchy is learned end-to-end via differentiable Adaptive Min-Cut Pooling rather than read off a fixed chemical decomposition, sidestepping vocabulary-coverage issues; and (ii) the framework supports an arbitrary number of scales over a single shared discrete codebook, decoded by a standard Transformer with no specialized tree decoder or motif-attachment module.

2.2 Hierarchical Graph Representations and Multi-Scale Generation

Differentiable graph pooling methods such as DiffPool [Ying et al., 2018] and MinCutPool [Bianchi et al., 2020] were originally designed for classification: their objectives reward node clustering but do not guarantee that the coarsened graph preserves the global topology needed for reconstruction. We repurpose MinCut pooling for generative compression by extracting every hierarchy level directly from the original graph (rather than recursively) and augmenting it with spectral regularization across scales. A complementary line of work imports next-scale prediction from image generation [Tian et al., 2024]: MAG [Belkadi et al., 2025] demonstrates scale-wise graph generation for inference acceleration but coarsens via implicit interpolation without topological constraints, limiting validity on complex molecules. Unlike motif-based hierarchical AR models such as HierVAE and MoLeR, H-GVT enforces topology preservation through spectral regularization. Its hierarchy levels are not labeled a priori as "motif" or "atom" but emerge from data-driven pooling, letting the model allocate its hierarchy budget where it minimizes reconstruction and spectral-consistency loss.

3 Methodology

We present the Hierarchical Graph VQ-Transformer (H-GVT), illustrated in Figure 1. Our framework addresses each of the three challenges identified in Section 1 with a specific technical component. Adaptive Min-Cut Pooling handles irregular graph sizes and structure-aware downsampling (Challenges 1–2), the Spectral Topology Consistency Loss prevents topological degeneration at intermediate scales (Challenge 3), and scale-ordered autoregressive generation enables efficient coarse-to-fine decoding (the speed–quality trade-off). The framework operates in two stages: Stage 1 compresses graphs into hierarchical discrete codes, and Stage 2 models their joint distribution autoregressively.

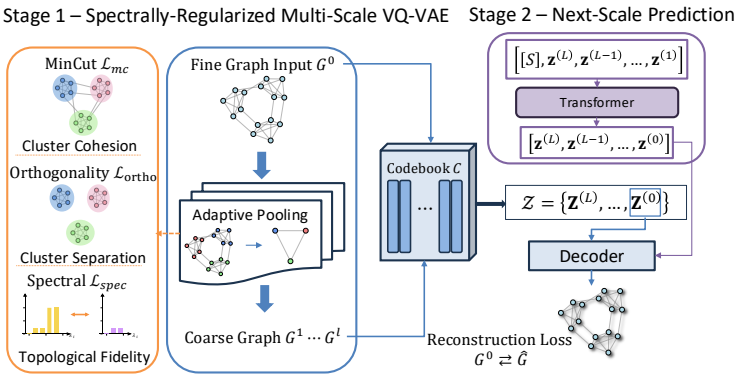

Figure 1: **Overview of the H-GVT framework. Stage 1 (Graph Compression):** The Spectrally-Regularized Multi-Scale VQ-VAE compresses the input molecular graph into hierarchical discrete codes via Adaptive Min-Cut Pooling, regularized by the Spectral Topology Consistency Loss. **Stage 2 (Coarse-to-Fine Generation):** A standard decoder-only Transformer generates tokens ordered from coarse to fine with level embeddings, first committing to global structural codes and then refining atomic details.

3.1 Spectrally-Regularized Multi-Scale VQ-VAE

Let $G = (V, E)$ be a molecular graph with N nodes, node features $\mathbf{H} \in \mathbb{R}^{N \times d_h}$, edge features $\mathbf{E} \in \mathbb{R}^{N \times N \times d_e}$, and adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. We define hierarchy levels $l \in \{0, 1, \ldots, L\}$, where l=0 is the original graph and l=L is the coarsest abstraction.

A Graph Transformer Encoder maps the input to continuous latent representations: $\mathbf{Z}^{(0)} = \text{GTE}(\mathbf{H}, \mathbf{E}) \in \mathbb{R}^{N \times d_{model}}$.

3.1.1 Adaptive Min-Cut Pooling with Dynamic Sizing

To obtain hierarchical representations, we compute coarsened feature matrices $\mathbf{Z}^{(l)} \in \mathbb{R}^{K_l \times d_{model}}$ and adjacency matrices $\mathbf{A}^{(l)} \in \mathbb{R}^{K_l \times K_l}$ for each scale $l \in \{1, \ldots, L\}$, where K_l is the target cluster count.

Each level l uses a scale-specific GNN to compute a soft assignment matrix $\mathbf{S}^{(l)} \in \mathbb{R}^{N \times K_l}$: $\mathbf{S}^{(l)} = \text{GNN}_l(\mathbf{Z}^{(0)}, \mathbf{A}^{(0)})$. The coarsening behaves differently depending on whether the graph size exceeds the target cluster count. When $N > K_l$, we apply dense MinCut pooling: $\mathbf{Z}^{(l)} = {\mathbf{S}^{(l)}}^\top \mathbf{Z}^{(0)}$ and $\mathbf{A}^{(l)} = {\mathbf{S}^{(l)}}^\top \mathbf{A}^{(0)} \mathbf{S}^{(l)}$. When $N \leq K_l$, no active pooling is needed, and we preserve the original information through zero-padding: $\mathbf{Z}^{(l)} = [\mathbf{Z}^{(0)} \parallel \mathbf{0}]$ and $\mathbf{A}^{(l)} = \text{Pad}(\mathbf{A}^{(0)})$.

Design rationale: parallel vs. recursive coarsening. Each $\mathbf{S}^{(l)}$ is computed from the original graph rather than recursively from level $l-1$, which avoids compounding approximation errors and grants each scale-specific GNN access to the full atom-level chemistry needed to inform coarse clustering; the Spectral Topology Consistency Loss (Eq. 3) further anchors every scale directly to the ground-truth topology. The hierarchy in H-GVT thus arises at generation time (Section 3.2), with fine-scale tokens conditioned on all coarser predecessors.

Cluster channel ordering. Stage 2 reads the pooled tokens in the channel order produced by GNN_l, without any per-graph permutation matching or sorting step. This raises an important question because the pooling losses are largely invariant to permutations of the columns of $\mathbf{S}^{(l)}$. In H-GVT, slot indexing is made reproducible by the combination of RCM-ordered atom tensors and fixed GNN output channels. The rows of $\mathbf{Z}^{(0)}$ follow the RCM atom order, which tends to place graph-adjacent atoms close in the sequence, while the columns of $\mathbf{S}^{(l)}$ are fixed output channels of a learned network rather than graph-specific cluster labels. Composed with this ordered input, the MinCut objective encourages each channel to specialize to a relatively localized, well-connected region of the RCM-indexed graph, and the orthogonality loss discourages different channels from covering the same region. We do not claim that channel j corresponds to a fixed chemical motif across all molecules. The claim is weaker and operational: the channel order defines a reproducible slot-indexed sequence layout, and this layout provides a structural inductive bias for the autoregressive prior. We

provide a more detailed mechanistic account in Appendix G. In addition to qualitative per-molecule visualizations, Appendix G.4 quantitatively verifies this interpretation by measuring the soft overlap between learned clusters and RDKit-detected ring systems and functional groups.

The pooling GNNs are trained via auxiliary MinCut and Orthogonality losses [Bianchi et al., 2020] computed on the original graph:

$$\mathcal{L}_{pool}^{(l)} = \underbrace{-\frac{\mathrm{Tr}(\mathbf{S}^{(l)\top}\mathbf{A}^{(0)}\mathbf{S}^{(l)})}{\mathrm{Tr}(\mathbf{S}^{(l)\top}\mathbf{D}^{(0)}\mathbf{S}^{(l)})}}_{\mathcal{L}_{mc}^{(l)}} + \underbrace{\left\| \frac{\mathbf{S}^{(l)\top}\mathbf{S}^{(l)}}{\|\mathbf{S}^{(l)\top}\mathbf{S}^{(l)}\|_F} - \frac{\mathbf{I}_{K_l}}{\sqrt{K_l}} \right\|_F}_{\mathcal{L}_{ortho}^{(l)}} \tag{1}$$

where $\mathbf{I}_{K_l}$ is the $K_l \times K_l$ identity matrix. This is the same normalized orthogonality penalty used by dense MinCut pooling, which encourages different cluster channels to be non-overlapping without introducing a scale mismatch across different K_l.

3.1.2 Spectral Topology Consistency Loss

A key limitation of standard pooling is the loss of global topological information during downsampling. We address this with a Spectral Topology Consistency Loss grounded in spectral graph theory: the smallest eigenvalues of the normalized Laplacian encode fundamental global properties such as algebraic connectivity, the number of connected components, and community structure [Fiedler, 1973, Chung, 1997, Von Luxburg, 2007, Bianchi et al., 2020]. Preserving these eigenvalues across scales thus provides a principled way to ensure that coarsened graphs retain the topological signatures of the original.

For a graph at scale l with adjacency $\mathbf{A}^{(l)}$ and degree matrix $\mathbf{D}^{(l)}$, the normalized Laplacian is:

$$\mathbf{L}^{(l)} = \mathbf{I} - (\mathbf{D}^{(l)})^{-1/2}\mathbf{A}^{(l)}(\mathbf{D}^{(l)})^{-1/2} \tag{2}$$

Let $\Lambda_k(\mathbf{L}^{(l)}) = \{\lambda_1^{(l)}, \ldots, \lambda_k^{(l)}\}$ denote the k smallest eigenvalues. For each pooled level $l \in \{1, \ldots, L\}$, we set $k_l = \lfloor K_l/2 \rfloor$ and minimize the spectral discrepancy with the original graph:

$$\mathcal{L}_{spec} = \sum_{l=1}^{L}\sum_{i=1}^{k_l} \left\| \lambda_i^{(0)} - \lambda_i^{(l)} \right\|^2 \tag{3}$$

For the passthrough case ($N \leq K_l$), masking leaves the active $N \times N$ graph (before padding) unchanged, so $\mathcal{L}_{spec} = 0$; thus the loss provides supervision only when active pooling is performed.

Remark 1. *While the Laplacian spectrum is not a complete invariant of graph isomorphism, matching its low-frequency components provides an effective and lightweight regularizer. On MOSES, including $\mathcal{L}_{spec}$ increases training time by only 2.8% (179s → 184s per epoch), while yielding a 3× reduction in NSPDK error on ZINC250k (Section 4.3).*

3.1.3 Multi-Scale Quantization and Decoding

We collect continuous representations from coarse to fine, $\mathcal{Z} = \{\mathbf{Z}^{(L)}, \ldots, \mathbf{Z}^{(0)}\}$, and discretize them using a shared codebook $\mathcal{C} \subset \mathbb{R}^{d_{model}}$, following vector-quantized autoencoders [Van Den Oord et al., 2017]. Each embedding is mapped to its nearest codebook vector: $\mathbf{z}_q^{(l)} = \arg\min_{\mathbf{e}\in\mathcal{C}} \|\mathbf{z}^{(l)} - \mathbf{e}\|_2$, with the Straight-Through Estimator for gradient propagation.

The decoder $\mathcal{D}$ takes the quantized fine-scale codes $\mathbf{Z}_q^{(0)}$ to reconstruct the molecular graph, following the GVT configuration [Zheng et al., 2026] with a Transformer Decoder [Vaswani et al., 2017] augmented with Rotary Positional Embeddings [Su et al., 2024]. The total Stage 1 objective is:

$$\mathcal{L}_{\text{Stage1}} = \underbrace{\lambda_{node}\mathcal{L}_{CE}(\mathbf{H}, \hat{\mathbf{H}}) + \lambda_{edge}\mathcal{L}_{CE}(\mathbf{E}, \hat{\mathbf{E}})}_{\mathcal{L}_{recon}} + \lambda_{commit}\mathcal{L}_{vq} + \lambda_p \sum_{l=1}^{L} \mathcal{L}_{pool}^{(l)} + \lambda_s \mathcal{L}_{spec} \tag{4}$$

How coarse codes acquire structural content. Although only the fine-scale codes $\mathbf{Z}_q^{(0)}$ are passed to the decoder for reconstruction, the coarse codes are not arbitrary auxiliary variables. Passing

only the fine-scale codes to the decoder is a deliberate design choice that separates reconstruction from the hierarchical components. A reconstruction ablation on QM9 shows that removing the spectral or pooling losses leaves Stage-1 node accuracy between 0.9996 and 0.9999 (Appendix B.1). The coarse codes are produced by MinCut-pooled, RCM-ordered atom representations, quantized through the same codebook $\mathcal{C}$ used by fine-scale codes, and regularized through $\mathcal{L}_{spec}$ to preserve low-frequency topological information. Thus, we interpret them as coarse structural context for Stage 2 generation rather than as directly decoded motifs. A more detailed account is given in Appendix G, and Appendix G.3 provides qualitative examples of coarse clusters aligned with recognizable substructures.

3.2 Scale-Ordered Autoregressive Generation

In Stage 2, we model the joint probability of the hierarchical discrete codes. The design is intentionally minimal: we use a standard decoder-only autoregressive Transformer [Vaswani et al., 2017] with a conventional causal mask, injecting hierarchy information solely through token ordering and level embeddings.

3.2.1 Hierarchical Sequence Construction

We flatten the multi-scale codes into a single sequence ordered from coarsest to finest: $\tau = [\mathbf{z}^{(L)}, \mathbf{z}^{(L-1)}, \ldots, \mathbf{z}^{(0)}]$. Each token's embedding is augmented with a level embedding:

$$h_i = \text{TokEmb}(t_i) + \text{PosEmb}(i) + \text{LevelEmb}(\ell_i) \tag{5}$$

where $\ell_i \in \{0, \ldots, L\}$ indicates the hierarchy level of token t_i. This allows the Transformer to distinguish between granularity levels without architectural modifications.

Causal masking. We use a standard causal mask rather than a scale-causal variant (bidirectional within levels). The causal mask already ensures that fine-scale tokens condition on all coarse-scale predecessors, which is exactly the dependency we need, while scale-causal masking introduces optimization difficulties that substantially hurt validity and FCD (Table 4; Appendix H). The joint probability therefore factorizes as $p(\tau) = \prod_t p(\tau_t \mid \tau_{<t})$, and applying this factorization to the scale-ordered sequence implicitly yields $p(\mathbf{z}^{(l)} \mid \mathbf{z}^{(>l)})$ without requiring any scale-specific factorization.

4 Experiments

We evaluate H-GVT on four standard molecular benchmarks: QM9 [Ramakrishnan et al., 2014, Wu et al., 2018], ZINC250k [Irwin et al., 2012], MOSES [Polykovskiy et al., 2020], and GuacaMol [Brown et al., 2019]. All molecules are preprocessed with RDKit [Bento et al., 2020] (hydrogen removal, Kekulization). We use standard data splits: 80%/20% for QM9 and ZINC250k, official scaffold splits for MOSES and GuacaMol. Stage-1 reconstruction accuracy exceeds 99.7% on all test sets (Appendix B), ensuring negligible information loss during quantization. Full hyperparameters are in Appendix C.

4.1 The Speed–Quality Frontier

We first address the central question of whether a single model can achieve both the structural fidelity of diffusion models and the speed of autoregressive models.

Figure 2 plots inference speed against generation quality (NSPDK) for all evaluated methods on QM9, benchmarked on a single NVIDIA RTX 4090 with official implementations and released checkpoints. We report wall-clock time to generate 10,000 molecules including RDKit construction.

Existing methods force a compromise: diffusion models (DiGress, SwinGNN) achieve low NSPDK but require $\sim$1,000s; flat AR models (DGAE) are fast but have high NSPDK error (1.5×10^{-3}). H-GVT closes this gap, generating 10,000 molecules in 14.3s (68$\times$ faster than DiGress) while simultaneously achieving the lowest NSPDK (2×10^{-4}) among all methods. This places H-GVT on the Pareto frontier, offering diffusion-level structural fidelity with AR-level efficiency. This claim concerns Pareto optimality on the speed–quality frontier and not the best value on every metric. SwinGNN, for example, obtains a lower FCD on QM9 (0.169 vs. 0.87) at 23$\times$ the sampling time (Table 1 and Figure 2).



Figure 2: **Speed–quality trade-off on QM9.** x-axis: generation time (log scale; left is faster). y-axis: NSPDK metric (log scale; bottom is better). **H-GVT (star)** occupies the Pareto-optimal bottom-left region: it achieves the lowest NSPDK (2×10^{-4}) while generating 10K molecules in 14.3s, 68× faster than DiGress and 23× faster than SwinGNN. Bubble size indicates validity.

Method	Class	QM9				ZINC250k			
		Val.↑	Uni.↑	FCD↓	NSPDK↓	Val.↑	Uni.↑	FCD↓	NSPDK↓
GraphAF [Shi et al., 2020]	Flow	57.16	83.78	5.384	2.10e-2	68.47	99.01	16.023	4.40e-2
GraphDF [Luo et al., 2021]		79.33	95.73	11.283	7.50e-2	41.84	93.75	40.51	3.54e-1
GDSS [Jo et al., 2022]	Diffusion	90.36	94.70	2.923	4.40e-3	97.35	99.76	11.398	1.80e-2
DiGress [Vignac et al., 2022]		95.43	93.78	0.643	7.28e-4	84.94	99.21	4.88	8.75e-3
SwinGNN [Yan et al., 2023]		99.68	95.92	**0.169**	4.02e-4	87.74	99.98	5.219	7.52e-3
GLAD [Boget et al., 2025]		97.12	97.52	0.201	3e-4	81.81	**100**	2.54	2.00e-3
DGAE [Boget et al., 2023]	Hybrid	92.0	97.61	0.86	1.50e-3	77.9	99.94	4.4	7.00e-3
GVT [Zheng et al., 2026]		99.76	95.94	0.87	6.0e-4	99.57	97.55	1.16	6.89e-2
PARD [Zhao et al., 2024]		97.5	95.8	-	-	95.23	99.99	1.98	-
GraphBFN [Xiong et al., 2025]	Multi-Scale	99.60	**99.97**	0.214	8e-4	96.00	**100**	5.743	6.9e-3
HDFM [Boget et al., 2026]		95.78	-	0.23	5.3e-4	99.30	-	1.23	4.3e-4
MAG [Belkadi et al., 2025]		89.43	83.14	3.84	-	87.88	84.22	19.70	-
H-GVT		**99.81**	95.36	0.87	**2e-4**	**99.86**	97.44	**1.147**	**1e-4**

Table 1: **Generation performance on QM9 and ZINC250k.** H-GVT achieves the best result on the majority of metrics. Its NSPDK score on ZINC250k (1×10^{-4}) is the lowest among all evaluated methods, including hierarchical denoising-based models (HDFM, GraphBFN) and diffusion baselines, confirming that coarse-to-fine autoregressive ordering substantially improves global structural coherence. All baseline numbers are taken as originally reported in the cited references.

4.2 Distribution Learning Benchmarks

4.2.1 Results on QM9 and ZINC250k

Table 1 compares H-GVT against strong baselines across model classes.

Global topology (NSPDK). H-GVT achieves NSPDK of 2×10^{-4} on QM9 and 1×10^{-4} on ZINC250k, surpassing all diffusion and hierarchical denoising models including GLAD (3×10^{-4} / 2×10^{-3}), GraphBFN (8×10^{-4} / 6.9×10^{-3}), and HDFM (5.3×10^{-4} / 4.3×10^{-4}), and is roughly 700× better than GVT on ZINC250k. This confirms that planning global structure at coarse scales before determining local details substantially improves topological coherence, even compared to recent denoising-based methods that also adopt hierarchical generation.

Distribution matching (FCD) and validity. H-GVT achieves the best FCD on ZINC250k (1.147) while maintaining near-perfect validity (99.86%). On QM9, FCD (0.87) is on par with the strongest AR baselines (DGAE 0.86, GVT 0.87) but trails the best diffusion models (SwinGNN 0.169, GLAD 0.201). We attribute this to QM9's small molecular size (≤ 9 heavy atoms): with so few atoms, a single coarse level often summarizes the entire molecule, so multi-scale factorization provides limited additional structure beyond what flat AR already captures. This is consistent with the observation that all three AR methods (DGAE, GVT, H-GVT) cluster tightly around FCD 0.87 on QM9, while diverging substantially on ZINC250k where molecules average 23 heavy atoms (H-GVT 1.147 vs.

Model	Class	Val.↑	Uni.↑	Nov.↑	FCD↓	SNN↑	Scaf↑
JT-VAE	Frag.	**100**	**100**	**99.9**	1.00	0.53	10.0
GraphINVENT	AR	96.4	99.8	-	1.22	0.54	<u>12.7</u>
DiGress	Diff.	85.7	**100**	<u>95.0</u>	1.19	0.52	**14.8**
PARD	AR+D	86.8	**100**	78.2	1.00	0.56	2.2
GVT†	AR	94.69	**100**	80.53	0.92	0.56	5.4
H-GVT	Ours	<u>98.5</u>	**100**	86.4	**0.19**	**0.59**	8.4

Table 2: **MOSES results.** †GVT reproduced under the Novelty-thresholded protocol (Nov. $\geq$ 80%; App. E). Under fair comparison, H-GVT achieves 4.8× better FCD (0.19 vs. 0.92) while maintaining higher Novelty and Scaffold diversity.

Model	Class	Val.↑	Uni.↑	Nov.↑	KL↑	FCD↑
NAGVAE	VAE	92.9	95.5	**100**	38.4	0.9
DiGress	Diff.	85.2	**100**	99.9	92.9	68.0
MolHIT	H-Diff.	87.1	**100**	98.73	96.7	54.9
GVT†	AR	92.17	99.9	92.57	96.45	73.24
H-GVT	Ours	**94.70**	**100**	92.60	**97.94**	**83.66**

Table 3: **GuacaMol results** (official scores; higher is better). †GVT reproduced under the Novelty-thresholded protocol (Nov. $\geq$ 90%; App. E).

DGAE 4.4 and GVT $\geq$1.16). The Pareto advantage of hierarchical generation thus manifests precisely where global topology becomes hierarchical.

4.2.2 Results on MOSES and GuacaMol

Fair comparison with GVT. The original GVT results [Zheng et al., 2026] reported unusually low Novelty (MOSES: 61.4%, GuacaMol: 82.4%), which suggests the model was evaluated in an overfitted regime where high distribution-matching scores (e.g., FCD 0.16) were achieved by memorizing training samples rather than learning the underlying distribution. To ensure a fair comparison, we reproduce GVT with the same Stage-2 Transformer capacity as H-GVT and apply a uniform checkpoint selection rule to both: we select the first checkpoint whose training-set Novelty exceeds a pre-set threshold (80% on MOSES, 90% on GuacaMol). The same rule is applied to H-GVT, so reported metric differences reflect modeling quality rather than different operating points on the memorization–fidelity curve. This reproduced GVT (denoted GVT†) serves as a Novelty-thresholded reproduced baseline. Appendix E empirically validates the protocol by showing that on GuacaMol, our largest benchmark, FCD and NSPDK improve by 3.0× and 3.2× respectively as outputs are progressively replaced with training molecules, confirming that both metrics reward memorization independently of modeling quality.

Resolving the fidelity–diversity trade-off (MOSES). Under this fair comparison, the advantage of hierarchical generation becomes clear. On MOSES, H-GVT achieves 4.8× better FCD (0.19 vs. 0.92) while simultaneously maintaining higher Novelty (86.4% vs. 80.5%), Validity (98.5% vs. 94.7%), and Scaffold diversity (8.4 vs. 5.4). These results show that coarse-to-fine generation does not merely redistribute the fidelity–diversity trade-off but shifts the Pareto frontier: H-GVT is better on both axes. We hypothesize that planning the molecular skeleton at coarse scales before determining local details lets the model explore a broader range of topological combinations, improving diversity without sacrificing distributional accuracy. We analyze this mechanism further in Section 4.4.

GuacaMol. The advantage is equally clear on the larger GuacaMol benchmark. At nearly identical Novelty (∼92.6%), H-GVT outperforms GVT† on every distribution metric: FCD improves from 73.24 to 83.66 (+14.2%), KL from 96.45 to 97.94, and Validity from 92.17% to 94.70%. It further surpasses the recent hierarchical-diffusion model MolHIT [Jung et al., 2026] on FCD (83.66 vs. 54.9) and KL (97.94 vs. 96.7), indicating that learned multi-scale graph compression provides a stronger structural prior than hand-designed hierarchical diffusion within the same hierarchy-aware paradigm. As shown in Appendix I, further scaling model capacity can yield even higher scores but at the cost of reduced Novelty, confirming the fidelity–diversity trade-off inherent in autoregressive models.

4.3 Ablation Studies

We conduct ablation studies on ZINC250k to isolate the contribution of each component. Table 4 includes the reproduced GVT† baseline (same Stage-2 capacity) to quantify the overall benefit of hierarchical generation.

Spectral Topology Consistency Loss. Removing $\mathcal{L}_{spec}$ increases NSPDK by 3× on ZINC250k ($1\times10^{-4} \rightarrow 3\times10^{-4}$) and degrades FCD (1.147 → 1.192), confirming that spectral regularization preserves global topology during coarsening. This improvement comes at minimal cost: only 2.8% additional training time on MOSES.

Variant	Val.↑	Uni.↑	FCD↓	NSPDK↓
GVT† (flat baseline)	98.99	97.94	1.166	6.8e-2
H-GVT (Full)	**99.86**	97.44	**1.147**	**1e-4**
w/o $\mathcal{L}_{spec}$	99.67	97.35	1.192	3e-4
w/o Adaptive Pooling	99.54	94.15	1.54	6e-4
Scale-Causal Mask	87.99	94.61	1.813	1.1e-3
Random Token Order	97.26	**99.99**	2.483	2.9e-3
w/o Level Embedding	99.23	97.69	1.174	**1e-4**

Table 4: **Ablation study on ZINC250k.** GVT† is the reproduced flat baseline with matched Stage-2 capacity. The full hierarchical framework yields a 680× NSPDK improvement over flat generation (1×10^{-4} vs. 6.8×10^{-2}). Among individual components, adaptive pooling has the largest impact on FCD ($1.147 \rightarrow 1.54$), spectral loss on NSPDK ($1\times10^{-4} \rightarrow 3\times10^{-4}$), and coarse-to-fine ordering is essential: random ordering degrades FCD by 2.2× and NSPDK by 29×. Scale-causal masking is actively harmful (validity drops to 87.99%).

Adaptive Min-Cut Pooling. Replacing our learned pooling with simple neighbor aggregation causes the largest degradation: FCD deteriorates to 1.54 and NSPDK increases 6× ($1\times10^{-4} \rightarrow 6\times10^{-4}$) on ZINC250k. This demonstrates that variable-sized molecular graphs cannot be effectively compressed with rigid pooling.

Standard vs. scale-causal masking. Replacing the standard causal mask with a scale-causal variant (bidirectional attention within coarse levels) dramatically hurts performance on ZINC250k: validity drops from 99.86% to 87.99%, FCD increases from 1.147 to 1.813, and NSPDK worsens by an order of magnitude. This validates our design choice: the hierarchy is encoded in the token ordering and level embeddings, making complex masking schemes not only unnecessary but actively harmful.

Coarse-to-fine token ordering. Shuffling the token order randomly across hierarchy levels (while retaining level embeddings to preserve scale identity) degrades FCD by 2.2× (1.147 → 2.483) and NSPDK by 29× ($1\times10^{-4} \rightarrow 2.9\times10^{-3}$), suggesting that the coarse-to-fine ordering itself, not merely the multi-scale codebook, is the critical mechanism enabling global structural planning. We analyze this finding in depth in Section 4.4.

Level embeddings. Removing the explicit level embeddings that signal scale identity causes only marginal degradation: FCD increases slightly (1.147 → 1.174) while NSPDK remains unchanged (1×10^{-4}). Contrasted with the severe failure of random ordering (29× NSPDK degradation despite retaining level embeddings), this asymmetric pair suggests that the positional structure of the coarse-to-fine ordering is largely sufficient on its own, and that explicit scale labels add only a modest signal beyond it. Level embeddings provide a small benefit to validity (99.86% → 99.23%) and are retained in the full model, but the bulk of the structural-planning effect comes from the ordering.

4.4 Why Coarse-to-Fine Ordering Matters

Three converging lines of evidence suggest that the coarse-to-fine ordering carries useful structural information. First, randomizing token order across levels while retaining level embeddings collapses NSPDK by 29× but preserves validity at 97.26% (Table 4), isolating ordering's role in structural planning from chemical correctness. Second, decoupled temperature control (impossible in flat AR) shows that T_1 governs global Novelty (recovering from $\sim$2% to $\sim$39% as T_1 grows from 0.5 to 1.5 on an overfitted model) while T_2 has only marginal effect on global metrics (Appendix J). Third, removing level embeddings while preserving ordering causes negligible degradation (NSPDK unchanged), whereas the converse (random order with level embeddings retained) is catastrophic, suggesting that the positional pattern of the coarse-to-fine ordering is largely sufficient on its own, with explicit scale labels contributing only a small additional signal. Further support is given by the single-molecule clustering case study and the quantitative chemical-alignment analysis in Appendix G.4, together with the property-level distributional analysis on GuacaMol in Appendix K.

4.5 Structural Content of the Coarse Codes

The ordering ablation cannot separate global structure in the coarse codes from easier sequence modeling, so we test the codes directly with two order-independent experiments.

Global target	Coarse codes	Fine codes	Shuffled control
Low-frequency Laplacian spectrum	**0.69**	0.46	0.00
Aromatic-ring count	**0.74**	0.73	0.00
Molecular size	**0.63**	0.50	0.00
Ring count	0.72	**0.76**	0.00
Fused-ring-system count	**0.58**	0.57	0.00
Bemis–Murcko scaffold (top-40, accuracy)	0.66	**0.69**	0.34

Table 5: **Linear probes on frozen Stage-1 codes** (60K ZINC250k molecules, held-out R^2, accuracy for the scaffold target).

Linear probes. We fit linear probes from the frozen coarse ($l \geq 1$) and fine ($l = 0$) codes to six global descriptors (Table 5). On the low-frequency Laplacian spectrum, the quantity optimized by $\mathcal{L}_{spec}$, the coarse codes obtain $R^2 = 0.69$ and the fine codes 0.46. Although the coarse codes summarize pooled clusters and not individual atoms, they also exceed the fine codes on aromatic-ring count, molecular size, and fused-ring-system count, and are slightly lower on ring count and scaffold identity. Combining both code sets gives the highest accuracy (ring count $R^2 = 0.85$), and no pooled level collapses to a small code vocabulary (Appendix G.5).

Causal intervention. To test whether the AR prior uses this information, we fix 300 coarse prefixes of real molecules and resample the fine level 16 times per prefix at temperature 1.3 (Appendix G.6). Of the completions, 46% are unique molecules, whereas 90.7% of the completions that share a prefix have the same ring count (ICC 0.88), and their ring count correlates with that of the source molecule at 0.97. The coarse prefix therefore determines most of the ring topology while the fine level varies the atom-level realization, which an order that only simplified sequence modeling would not produce.

4.6 Additional Analyses

Property-conditioned generation. With Stage 1 frozen, we add QED and $\log P$ decile embeddings to Eq. (5) and retrain only the Stage-2 prior on ZINC250k. The realized properties follow the targets with Pearson $r = 0.966$ (QED) and $r = 0.988$ ($\log P$) at 88.5–98.6% validity and 100% uniqueness (Appendix L).

Further results. Appendices A.5, B.1, F.1, K.1, and M report the training cost, a reconstruction ablation, the solver cost and gradient stability of $\mathcal{L}_{spec}$, structural-alert pass rates and scaffold novelty, and the effect of molecule size.

5 Conclusion and Limitations

We presented H-GVT, a hierarchical graph generation framework that achieves a new Pareto-optimal point on the speed–quality frontier for molecular graph generation. By compressing molecular graphs into topology-aware hierarchical discrete codes and generating them in coarse-to-fine order, H-GVT achieves best-in-comparison NSPDK on QM9 and ZINC250k, while improving fidelity–diversity trade-offs on MOSES and GuacaMol. The Spectral Topology Consistency Loss provides a principled, lightweight regularizer that preserves global topological invariants during coarsening. Linear probes and a causal intervention (Section 4.5) show that the coarse codes encode global topology that the AR prior uses during sampling, and a frozen tokenizer supports property-conditioned generation (Section 4.6).

Limitations. Spectral regularization is a strong but non-unique prior on topology, and the $O(N^3)$ eigen-decomposition would require sparse approximations on much larger graphs. H-GVT also focuses on 2D molecular graphs; extending to 3D geometry, scaling the property-conditioned generation of Section 4.6 beyond a feasibility study, and adaptive selection of hierarchy levels across chemical domains are promising future directions. At $N = 10^4$, a sparse top-k solver is 6–17$\times$ faster (Appendix F.1) and gives numerically identical gradients on molecular graphs (Appendix F.2). Stage 1 accounts for 81–96% of the training cost (Appendix A.5).

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China [62071154]. The authors declare no competing interests.

References

Viraj Bagal, Rishal Aggarwal, PK Vinod, and U Deva Priyakumar. Molgpt: molecular generation using a transformer-decoder model. *Journal of chemical information and modeling*, 62(9):2064–2076, 2021.

Samuel Belkadi, Steve Hong, Marian Chen, Miruna Cretu, Charles Harris, and Pietro Lio. Diffusion-free graph generation with next-scale prediction. *arXiv preprint arXiv:2503.23612*, 2025.

A Patrícia Bento, Anne Hersey, Eloy Félix, Greg Landrum, Anna Gaulton, Francis Atkinson, Louisa J Bellis, Marleen De Veij, and Andrew R Leach. An open source chemical structure curation pipeline using rdkit. *Journal of cheminformatics*, 12(1):51, 2020.

Steven H Bertz. The first general index of molecular complexity. *Journal of the American Chemical Society*, 103(12):3599–3601, 1981.

Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. In *International conference on machine learning*, pages 874–883. PMLR, 2020.

G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

Yoann Boget, Magda Gregorova, and Alexandros Kalousis. Discrete graph auto-encoder. *arXiv preprint arXiv:2306.07735*, 2023.

Yoann Boget, Frantzeska Lavda, Alexandros Kalousis, et al. Glad: Improving latent graph generative modeling with simple quantization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19695–19702, 2025.

Yoann Boget, Pablo Strasser, and Alexandros Kalousis. Hierarchical discrete flow matching for graph generation. *arXiv preprint arXiv:2604.00236*, 2026.

Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3): 1096–1108, 2019.

Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

Fabrizio Costa and Kurt De Grave. Fast neighborhood subgraph pairwise distance kernel. In *Proceedings of the 26th International Conference on Machine Learning*, pages 255–262. Omnipress; Madison, WI, USA, 2010.

Elizabeth Cuthill and James McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*, pages 157–172, 1969.

Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):8, 2009.

Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2): 298–305, 1973.

Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. Zinc: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7): 1757–1768, 2012.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pages 2323–2332. PMLR, 2018.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In *International conference on machine learning*, pages 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 *International conference on machine learning*, pages 10362–10383. PMLR, 2022.

Hojung Jung, Rodrigo Hormazabal, Jaehyeong Jo, Youngrok Park, Kyunggeun Roh, Se-Young Yun, Sehui Han, and Dae-Woong Jeong. Molhit: Advancing molecular-graph generation with hierarchical discrete diffusion models. *arXiv preprint arXiv:2602.17602*, 2026.

Youngchun Kwon, Jiho Yoo, Youn-Suk Choi, Won-Joon Son, Dongseon Lee, and Seokho Kang. Efficient learning of non-autoregressive graph variational autoencoders for molecular graph generation. *Journal of Cheminformatics*, 11(1):70, 2019.

Youngchun Kwon, Dongseon Lee, Youn-Suk Choi, Kyoham Shin, and Seokho Kang. Compressed graph representation for scalable molecular graph generation. *Journal of Cheminformatics*, 12(1): 58, 2020.

Yibo Li, Liangren Zhang, and Zhenming Liu. Multi-objective de novo drug design with conditional graph generative model. *Journal of cheminformatics*, 10(1):33, 2018.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Will Hamilton, David K Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. *Advances in neural information processing systems*, 32, 2019.

Youzhi Luo, Keqiang Yan, and Shuiwang Ji. Graphdf: A discrete flow model for molecular graph generation. In *International conference on machine learning*, pages 7192–7203. PMLR, 2021.

Krzysztof Maziarz, Henry Jackson-Flux, Pashmina Cameron, Finton Sirockin, Nadine Schneider, Nikolaus Stiefl, Marwin Segler, and Marc Brockschmidt. Learning to extend molecular scaffolds with structural motifs. *arXiv preprint arXiv:2103.03864*, 2021.

Rocío Mercado, Tobias Rastemo, Edvard Lindelöf, Günter Klambauer, Ola Engkvist, Hongming Chen, and Esben Jannik Bjerrum. Graph networks for molecular design. *Machine Learning: Science and Technology*, 2(2):025023, 2021.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, et al. Molecular sets (moses): a benchmarking platform for molecular generation models. *Frontiers in pharmacology*, 11:565644, 2020.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Gunter Klambauer. Fréchet chemnet distance: a metric for generative models for molecules in drug discovery. *Journal of chemical information and modeling*, 58(9):1736–1741, 2018.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Bidisha Samanta, Abir De, Gourhari Jana, Vicenç Gómez, Pratim Chattaraj, Niloy Ganguly, and Manuel Gomez-Rodriguez. Nevae: A deep generative model for molecular graphs. *Journal of machine learning research*, 21(114):1–33, 2020.

Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, 4(1): 120–131, 2018.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*, 2020.

Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *International conference on artificial neural networks*, pages 412–422. Springer, 2018.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Yida Xiong, Jiameng Chen, Kun Li, Hongzhi Zhang, Xiantao Cai, and Wenbin Hu. Hierarchical bayesian flow networks for molecular graph generation. *arXiv preprint arXiv:2510.10211*, 2025.

Qi Yan, Zhengyang Liang, Yang Song, Renjie Liao, and Lele Wang. Swingnn: Rethinking permutation invariance in diffusion models for graph generation. *arXiv preprint arXiv:2307.01646*, 2023.

Ling Yang, Ye Tian, Minkai Xu, Zhongyi Liu, Shenda Hong, Wei Qu, Wentao Zhang, Bin Cui, Muhan Zhang, and Jure Leskovec. Vqgraph: Rethinking graph representation space for bridging gnns and mlps. *arXiv preprint arXiv:2308.02117*, 2023.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31, 2018.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International conference on machine learning*, pages 5708–5717. PMLR, 2018.

Lingxiao Zhao, Xueying Ding, and Leman Akoglu. Pard: Permutation-invariant autoregressive diffusion for graph generation. *Advances in Neural Information Processing Systems*, 37:7156–7184, 2024.

Haozhuo Zheng, Cheng Wang, and Yang Liu. Graph vq-transformer (gvt): Fast and accurate molecular generation via high-fidelity discrete latents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(34):28804–28811, Mar. 2026. doi: 10.1609/aaai.v40i34.40114. URL `https://ojs.aaai.org/index.php/AAAI/article/view/40114`.

A Reproducibility Details

A.1 Dataset Preprocessing and Feature Encoding

Starting from the raw SMILES strings, we use RDKit to parse each SMILES into a molecular object. To standardize the graph representation, we Kekulize each molecule to resolve aromatic systems into explicit alternating single/double bonds, and remove all hydrogen atoms. The final molecular graph is constructed with atoms as nodes and explicit bonds as edges. Following GVT [Zheng et al., 2026], we additionally apply Reverse Cuthill–McKee (RCM) ordering [Cuthill and McKee, 1969] to the atom indices of each molecular graph after preprocessing. RCM provides a deterministic bandwidth-reducing ordering in which graph-adjacent atoms tend to remain close in sequence. This atom order is preserved throughout Stage-1 encoding and Stage-2 sequence construction, and underlies the slot-ordering discussion in Section 3.

A.2 Dataset Statistics

Dataset	#Mol.	#Train	#Test	Atom Count (Heavy Atoms)			
				Min	Median	Mean	Max
QM9	129012	116111	12901	1	9	8.80	9
ZINC250k	237679	190156	47523	6	23	23.11	38
MOSES	1760737	1584663	176074	8	22	21.64	27
GuacaMol	1427627	1202295	225332	2	27	27.94	88

Table 6: Dataset sizes/splits and atom count statistics after preprocessing.

Atom and bond vocabularies. For each dataset, we define a discrete vocabulary of atom types (e.g., {C, N, O, F} for QM9) and bond types (Single, Double, Triple). Both node and edge features are encoded as one-hot vectors according to their vocabularies. For example, if the atom vocabulary order is (C, N, O, F), then oxygen is encoded as $[0, 0, 1, 0]$. This yields node feature matrices $\mathbf{H} \in \mathbb{R}^{N \times d_h}$ and edge feature tensors $\mathbf{E} \in \mathbb{R}^{N \times N \times d_e}$ used by our model.

Implementation notes. We keep only the explicit chemical graph after preprocessing (atoms and bonds), and do not use additional handcrafted descriptors. Any SMILES that cannot be successfully parsed or Kekulized by RDKit are discarded to ensure a consistent graph construction.

A.3 Software and Hardware Environment

All model training and large-scale benchmark evaluations were conducted on a server running Ubuntu 22.04 with four NVIDIA RTX 5090 GPUs (32GB VRAM each). Inference speed benchmarks were performed separately on a single NVIDIA RTX 4090 GPU to ensure fair and consistent comparison with prior work.

Our implementation is based on PyTorch and PyTorch Geometric. Key library versions are: PyTorch 2.8.0 (CUDA 12.8), PyTorch Geometric 2.7.0, and RDKit 2024.9.6. For benchmark evaluation, we use the official GuacaMol implementation (guacamol 0.5.5) and the MOSES toolkit (molsets 1.0).

A.4 Evaluation Protocol and Metrics

Evaluation protocol. For H-GVT and reproduced GVT baselines, we generate 10,000 molecules per run and repeat sampling three times. For other published baselines, we report the numbers from the original papers, as their released implementations, checkpoints and evaluation protocols are not uniformly available.

Metrics. To evaluate molecular generation quality and distribution matching, we adopt widely-used metrics from standard benchmarks such as GuacaMol [Brown et al., 2019] and MOSES [Polykovskiy et al., 2020]. Core metrics include Validity (Val.), Uniqueness (Uni.), and Novelty, which jointly assess chemical correctness and diversity; the Fréchet ChemNet Distance (FCD) [Preuer et al., 2018],

which measures distribution similarity in a learned chemical feature space (for MOSES/ZINC/QM9 we report the raw FCD distance, where lower is better; for GuacaMol distribution-learning we follow the official implementation and report the GuacaMol FCD score variant, where higher is better); and NSPDK [Costa and De Grave, 2010], which measures structural distribution discrepancy based on the Neighborhood Subgraph Pairwise Distance Kernel (lower is better). For MOSES, we additionally report scaffold-level metrics such as SNN and Scaf. For GuacaMol distribution-learning, we also report the benchmark-defined KL divergence (KL div) score, which compares physicochemical property distributions (higher is better).

Validity. Validity measures the percentage of generated molecules that can be parsed into chemically valid structures. Concretely, a generated sample is counted as valid if RDKit can successfully construct and sanitize the molecule (e.g., passing valence and aromaticity checks). We compute

$$\text{Val.} = \frac{\#\{\text{valid molecules}\}}{\#\{\text{generated molecules}\}} \times 100\%.$$

Uniqueness. Uniqueness measures the fraction of distinct structures among the valid molecules, and is commonly used to detect mode collapse. We canonicalize each valid molecule into a canonical SMILES (via RDKit canonicalization) and count duplicates. Formally,

$$\text{Uni.} = \frac{\#\{\text{unique valid molecules}\}}{\#\{\text{valid molecules}\}} \times 100\%.$$

Novelty. Novelty measures the fraction of valid unique molecules that do not appear in the training set. Following common practice [Vignac et al., 2022], we compute novelty by membership checking on canonical SMILES:

$$\text{Nov.} = \frac{\#\{\text{valid unique molecules not in training set}\}}{\#\{\text{valid unique molecules}\}} \times 100\%.$$

Distribution metrics. FCD evaluates the Fréchet distance between generated and reference molecules in the ChemNet embedding space (lower indicates closer distributions). NSPDK evaluates discrepancies in substructure statistics via a graph kernel-based distance (lower is better). We follow the standard benchmark implementations for MOSES and GuacaMol for scaffold- and property-level metrics.

A.5 Training Cost

Table 7 reports the wall-clock training time of both stages on NVIDIA RTX 5090 GPUs. Stage 1 is trained for 2000 epochs on every dataset.

	Stage-1 VQ-VAE			Stage-2 AR prior				
Dataset	GPUs	Time	GPU-h	GPUs	Time	GPU-h	Total GPU-h	Stage-1 share
ZINC250k	4	1 d 13 h	148	2	3 h 22 min	6.7	154.7	95.7%
MOSES	4	4 d 7 h	412	4	1 d 0 h 15 min	97	509	80.9%
GuacaMol	4	15 d	1440	4	1 d 3 h	108	1548	93.0%

Table 7: Training cost of H-GVT on NVIDIA RTX 5090 GPUs.

Stage 1 accounts for 81–96% of the total training cost. It is trained once per dataset and then frozen, and one frozen tokenizer serves several Stage-2 priors. The property-conditioned prior of Appendix L reuses the frozen ZINC250k tokenizer and is trained in 1 h 50 min on a single GPU. Three of the variants in Table 4 (token order, level embeddings, and causal mask) change only Stage 2 and require hours of training instead of days. For a new conditioning task or a prior-side ablation, the additional cost is therefore the Stage-2 cost and not the total cost.

B Stage-1 Reconstruction Fidelity

A basic prerequisite for our two-stage framework is that the Stage-1 VQ-VAE must compress molecular graphs into discrete codes without losing important structural information. To verify this,

we evaluated the reconstruction accuracy of the pre-trained Stage-1 models on the held-out test sets of all four benchmarks. Reconstruction accuracy is calculated as the percentage of input graphs that are perfectly isomorphic to the graphs decoded from their latent discrete representations.

As shown in Table 8, H-GVT achieves near-perfect reconstruction rates across all datasets ($> 99.7\%$ on test sets). This indicates that our Spectrally-Regularized Multi-Scale VQ-VAE, combined with Adaptive Min-Cut Pooling, reduces the quantization error often observed in discrete latent models. The high fidelity of Stage-1 ensures that the generation quality of the Stage-2 Transformer is not bottlenecked by the expressiveness of the discrete representation.

Dataset	**Train Recon. (%)**	**Test Recon. (%)**
QM9	100.00	99.98
ZINC250k	99.99	99.84
MOSES	99.99	99.97
GuacaMol	99.96	99.78

Table 8: Stage-1 VQ-VAE reconstruction accuracy (percentage of perfectly reconstructed molecules).

B.1 Reconstruction Ablation

The Stage-1 decoder receives only the fine-scale codes $\mathbf{Z}_q^{(0)}$. This design isolates the reconstruction pathway from the hierarchical components. Reconstruction is carried by the fine-scale tokenizer, and the pooling and spectral losses shape the coarse codes without being traded off against reconstruction. Under this design, the hierarchical components are not expected to affect reconstruction. We test this prediction with four Stage-1 variants on QM9 trained under a matched short budget (200 epochs, batch size 1024, seed 42). These are controlled ablation runs and not the 2000-epoch models of Table 8, so their absolute perfect-reconstruction rates are much lower and should be compared only with each other. A graph counts as perfectly reconstructed only if all its nodes and edges are recovered exactly, so this rate compounds over all $O(N^2)$ edge decisions.

Variant	Train perfect	Test perfect	Node acc.	Edge acc.
Full	0.729	0.728	0.9996	0.9884
No spectral loss (λ_s=0)	0.718	0.719	0.9996	0.9879
No pooling and spectral losses (λ_p=λ_s=0)	0.759	0.759	0.9996	0.9893
Static pooling	0.772	0.771	0.9999	0.9972

Table 9: Stage-1 reconstruction ablation on QM9 (200 epochs, batch size 1024, seed 42). Perfect reconstruction is the fraction of graphs whose nodes and edges are all recovered exactly.

Node accuracy is between 0.9996 and 0.9999 for all variants, and edge accuracy differs by less than one percentage point (0.9879 to 0.9972). Train and test perfect-reconstruction rates differ by at most 0.001 for every variant, so the results do not indicate memorization. We attribute the remaining differences in perfect reconstruction (0.728 for the full model vs. 0.771 for static pooling) to incomplete optimization under the 200-epoch budget. With the full training budget, the full model reaches 99.98% test reconstruction on QM9 (Table 8). The spectral loss and the pooling components do not affect reconstruction but change generation quality substantially (Table 4), so their effect on generation is not mediated by reconstruction.

C Hyperparameters

We summarize the hyperparameters in Table 10 for both the Stage-1 Graph VQ-VAE and the Stage-2 autoregressive Transformer. For Stage 1, the architecture captures multi-scale topological features through a series of hierarchical pooling layers, with hyperparameters tuned to balance reconstruction fidelity and latent space regularity. For Stage 2, we use a Transformer-based prior to model the hierarchical discrete codes. This stage uses a unified architecture across all benchmarks to evaluate the robustness of our scale-ordered generation. During inference, we apply temperature scaling and Top-k sampling to support diverse molecular generation while maintaining high chemical validity.

Parameter	GuacaMol	MOSES	ZINC250k	QM9
Stage-1 Model Architecture (VQ-VAE)				
Encoder #Layers	8	8	8	8
Encoder Hidden Dim	192	128	192	192
Encoder Output Dim	64	64	48	32
Encoder Node Feat Dim	12	7	9	4
PE Dim	6	6	6	6
Decoder #Layers	8	8	8	8
Decoder Hidden Dim	192	128	192	192
Decoder Input Dim	64	64	48	32
Decoder Output Dim	12	7	9	4
Codebook Size	4096	1024	4096	512
VQ Decay	0.99	0.99	0.99	0.99
Cluster Sizes (K_l)	[32,16,8,4,2,1]	[16,8,4,2,1]	[16,8,4,2,1]	[4,2,1]
Dropout Rate	0.1	0.1	0.1	0.1
Stage-1 Loss Weights & Training				
Node Loss (λ_{node})	1.0	1.0	1.0	1.0
Edge Loss (λ_{edge})	1.0	1.0	1.0	1.0
Pooling Loss (λ_{p})	0.1	0.1	0.1	0.1
Spectral Loss (λ_{s})	0.1	0.1	0.1	0.1
VQ Loss (λ_{commit})	0.2	0.2	0.2	0.2
Batch Size	64	512	384	256
Learning Rate	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
Stage-2 Transformer (Prior)				
Transformer #Layers	12	12	12	12
Embedding Dim (n_{embd})	640	512	512	512
Attention Heads (n_{head})	16	16	16	16
Attention Dropout	0.1	0.1	0.1	0.1
Embedding Dropout	0.1	0.1	0.1	0.1
Residual Dropout	0.1	0.1	0.1	0.1
Inference / Sampling				
Temperature (τ)	1.0	1.0	1.0	1.0
Top-k	50	50	50	50

Table 10: Hyperparameters for Stage-1 (H-GVT), Stage-2 (Prior), and Inference across different datasets.

D Algorithmic Details

The training of H-GVT follows a decoupled paradigm: Stage 1 optimizes a discrete hierarchical latent space with explicit multi-scale topological regularization, while Stage 2 learns a scale-ordered autoregressive prior over the resulting codes. Algorithm 1 details the spectrally-regularized multi-scale VQ-VAE training that jointly enforces pooling quality and spectral topology consistency across scales. Algorithm 2 then trains a Transformer to model the coarse-to-fine code sequence. Finally, Algorithm 3 summarizes the hierarchical sampling procedure, generating latent codes from coarse to fine and decoding them into a molecular graph.

D.1 Sequence Layout and Padding Protocol

We provide precise details of how the multi-scale discrete codes are arranged into a single sequence for Stage-2 training and inference.

Fixed-length coarse tokens. For each hierarchy level $l \in \{1, \dots, L\}$, the token count is fixed at K_l (the target cluster count for that level). When a molecule has fewer atoms than K_l (i.e., $N \leq K_l$), the passthrough case (Section 3) produces zero-padded representations that are subsequently quantized through the shared VQ codebook into standard codebook indices. These tokens are not treated specially: they participate in the autoregressive loss and are generated during inference, ensuring a

Algorithm 1 Stage 1: Spectrally-Regularized Multi-Scale VQ-VAE Training

Require: Molecular graphs $\mathcal{G}$, cluster sizes $\{K_l\}_{l=1}^{L}$, weights $\lambda_{edge}, \lambda_{commit}, \lambda_p, \lambda_s$
Ensure: Encoder $\mathcal{E}$, Pooling layers $\{\mathcal{P}_l\}$, Codebook $\mathcal{C}$, Decoder $\mathcal{D}$

while not converged **do**
  Sample batch $G = (V, E) \in \mathcal{G}$
  $\mathbf{Z}^{(0)}, \mathbf{A}^{(0)} \leftarrow \mathcal{E}(G)$
  $\mathcal{L}_{spec} \leftarrow 0, \mathcal{L}_{pool} \leftarrow 0, \mathcal{Z}_{multi} \leftarrow \emptyset$
  for $l = 1$ **to** L **do**
    $\mathbf{Z}^{(l)}, \mathbf{A}^{(l)}, \ell_{mc}, \ell_{ortho} \leftarrow \mathcal{P}_l(\mathbf{Z}^{(0)}, \mathbf{A}^{(0)})$
    $\mathcal{L}_{pool} \leftarrow \mathcal{L}_{pool} + \ell_{mc} + \ell_{ortho}$
    if $N > K_l$ **then**
      $\mathcal{L}_{spec} \leftarrow \mathcal{L}_{spec} + \text{SpectralLoss}(\mathbf{A}^{(0)}, \mathbf{A}^{(l)}, k = \lfloor K_l/2 \rfloor)$
    else
      $\mathcal{L}_{spec} \leftarrow \mathcal{L}_{spec} + 0$
    end if
    $\mathcal{Z}_{multi} \leftarrow \mathcal{Z}_{multi} \cup \{\mathbf{Z}^{(l)}\}$
  end for
  $\hat{\mathbf{Z}}_{hier}, \mathcal{L}_{vq_h} \leftarrow \text{VQ}(\text{Concat}(\mathcal{Z}_{multi}), \mathcal{C})$
  $\hat{\mathbf{Z}}^{(0)}, \mathcal{L}_{vq_f} \leftarrow \text{VQ}(\mathbf{Z}^{(0)}, \mathcal{C})$
  $\hat{V}, \hat{E} \leftarrow \mathcal{D}(\hat{\mathbf{Z}}^{(0)})$
  $\mathcal{L} \leftarrow \mathcal{L}_{recon} + \lambda_{commit}(\mathcal{L}_{vq_h} + \mathcal{L}_{vq_f}) + \lambda_p \mathcal{L}_{pool} + \lambda_s \mathcal{L}_{spec}$
  Update all parameters
end while

Algorithm 2 Stage 2: Scale-Ordered Autoregressive Training

Require: Discrete codes $\{\mathbf{z}_i\}$ from Stage 1
Ensure: Transformer parameters θ_{AR}

while not converged **do**
  Sample codes $\mathbf{z} = \{z_0, z_1, \ldots, z_L\}$
  $\tau_{src} \leftarrow [\text{[SOS]}, z_L, z_{L-1}, \ldots, z_0]$
  $\tau_{tgt} \leftarrow [z_L, z_{L-1}, \ldots, z_0, \text{[EOS]}]$
  $\ell \leftarrow [L, \underbrace{L, \ldots, L}_{K_L}, \underbrace{L-1, \ldots, L-1}_{K_{L-1}}, \ldots, \underbrace{0, \ldots, 0}_{N}]$
  $\mathbf{H} \leftarrow \text{TokEmb}(\tau_{src}) + \text{PosEmb} + \text{LevelEmb}(\ell)$
  $\hat{\mathbf{P}} \leftarrow \text{Transformer}_{\theta_{AR}}(\mathbf{H})$
  $\mathcal{L}_{AR} \leftarrow \text{CrossEntropy}(\hat{\mathbf{P}}, \tau_{tgt})$
  Update θ_{AR} via AdamW
end while

uniform sequence layout across molecules of varying sizes. Within each level, tokens are ordered by the fixed output-channel index of GNN_l; we do not perform molecule-specific cluster sorting or permutation matching.

Variable-length fine tokens. The fine-scale tokens (level l=0) correspond to individual atoms and thus vary in count across molecules. During training, the fine-scale subsequence is terminated by an `[EOS]` token. During inference, generation of fine-scale tokens proceeds autoregressively until either `[EOS]` is sampled or the sequence length reaches N_{max}.

Batch padding and loss masking. Within a training batch, sequences of different lengths are right-padded to a uniform length using a dedicated `[PAD]` token (vocabulary index $|\mathcal{C}|+1$, distinct from any codebook entry). The cross-entropy loss is computed only over non-`[PAD]` positions; coarse-level zero-padded tokens (which carry valid codebook indices) are not masked.

N_{max} **determination.** The maximum fine-scale sequence length N_{max} is derived deterministically from the Stage-1 checkpoint: it equals the maximum atom count observed during Stage-1 VQ

Algorithm 3 Coarse-to-Fine Molecular Sampling

Require: Transformer θ_{AR}, Decoder $\mathcal{D}$, Codebook $\mathcal{C}$
Ensure: Generated molecular graph $\hat{G}$
$\tau \leftarrow$ [[SOS]]
$\tau_{\text{fine}} \leftarrow [\,]$
for $l = L$ **to** 1 **do**
  for $j = 1$ **to** K_l **do**
    $t \sim P(\cdot \mid \tau, \text{LevelEmb}(l); \theta_{AR})$
    Append t to τ
  end for
end for
repeat
  $t \sim P(\cdot \mid \tau, \text{LevelEmb}(0); \theta_{AR})$
  Append t to τ
  if $t \neq$ [EOS] **then**
    Append t to τ_{fine}
  end if
until $t =$ [EOS] **or** length exceeds N_{max}
$\hat{\mathbf{Z}}^{(0)} \leftarrow \text{LookUp}(\tau_{\text{fine}}, \mathcal{C})$
$\hat{V}, \hat{E} \leftarrow \mathcal{D}(\hat{\mathbf{Z}}^{(0)})$
return $\hat{G} \leftarrow \text{ConstructMolecular}(\hat{V}, \hat{E})$

encoding over the training set. This value is persisted alongside the Stage-1 checkpoint and used identically during both Stage-2 training and inference, ensuring strict consistency.

E Fair Comparison Protocol for GVT†

This appendix substantiates the Novelty-thresholded protocol introduced in Section 4.2. We first quantify, on the largest benchmark in our study, how much of the advantage attributed to distribution-matching metrics is in fact attributable to memorization. We then document the checkpoint-selection rule applied uniformly to GVT† and H-GVT, and present the original GVT, GVT†, and H-GVT numbers side-by-side.

E.1 The Novelty–Fidelity Confound on GuacaMol

FCD and NSPDK are the primary distribution-matching metrics used in molecular generation benchmarks. Both compare a set of generated molecules against a reference set using population-level statistics (Fréchet distance over ChemNet activations for FCD, MMD over NSPDK graph kernels for NSPDK), which makes them inherently rewarded when generated outputs are drawn from, or close to, the training distribution. Figure 3 quantifies this bias on GuacaMol, the largest dataset in our study: pure training-set retrieval (x=1) achieves FCD $= 0.37$ and NSPDK $= 1.5\times10^{-4}$, numbers that reflect memorization rather than generative capability. Importantly, this confound is not an artifact of small datasets: even with GuacaMol's ≈1.5M training molecules, the $3\times$ metric swing persists across the full Novelty range. Any comparison that does not control for Novelty therefore conflates two distinct quantities: how well the model captures the data distribution, and how closely it operates to training-set retrieval.

E.2 GVT† Reproduction Protocol

What is shared with original GVT. GVT† uses the architecture, hyperparameters, and training recipe of the original GVT [Zheng et al., 2026] unchanged. We reproduce training end-to-end on MOSES and GuacaMol using the same Stage-1 VQ-VAE and Stage-2 Transformer configurations. Stage-2 capacity is additionally matched to H-GVT's to ensure that comparisons isolate the hierarchical-generation contribution rather than parameter count.

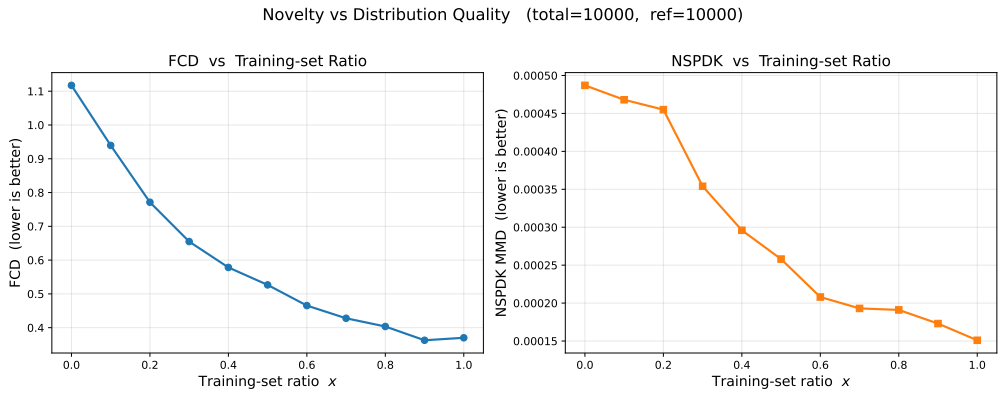

Figure 3: **Distribution-matching metrics reward memorization even at scale.** On the GuacaMol benchmark (training set ≈1.5M molecules; 10,000 generated samples vs. 10,000 reference), we progressively replace a fraction x of generated outputs with training molecules and recompute FCD and NSPDK MMD. Both metrics improve monotonically as x increases: FCD drops from 1.117 (x=0, 100% novel) to 0.370 (x=1, pure training-set retrieval), a 3.0× improvement; NSPDK drops from 4.87×10^{-4} to 1.51×10^{-4}, a 3.2× improvement. Any model that sacrifices Novelty therefore scores better on these metrics by construction, independent of its modeling quality. Fair comparison requires matched Novelty.

Table 11: **MOSES: Original GVT vs. GVT† vs. H-GVT** (FCD lower is better). Original GVT reports FCD 0.16 at Novelty 61.4%; Figure 3 shows that a $\sim 3\times$ FCD advantage is attainable purely by reducing Novelty, so this number is not directly comparable to baselines operating in the 80–95% Novelty range. Under the Novelty-thresholded protocol (Nov. $\geq$ 80%), GVT† vs. H-GVT isolates the architectural contribution of hierarchical generation.

Model	Val.↑	Uni.↑	Nov.↑	FCD↓	SNN↑	Scaf↑
GVT (original) [Zheng et al., 2026]	99.42	100.0	61.43	0.16	0.64	3.4
GVT† (ours, Nov. $\geq$ 80%)	94.69	100.0	80.53	0.92	0.56	5.4
H-GVT (same rule)	98.5	100.0	86.4	0.19	0.59	8.4

What differs: the checkpoint selection rule. The only departure from the original GVT protocol is checkpoint selection. Rather than selecting the checkpoint with the best FCD, which Figure 3 shows is biased toward low-Novelty solutions, we apply a Novelty-thresholded early-stopping rule. During training, we monitor training-set Novelty at each evaluation epoch and select the first checkpoint whose Novelty exceeds a pre-set threshold (80% on MOSES, 90% on GuacaMol). The same rule is applied to H-GVT. This ensures both models are evaluated at matched operating points on the memorization–fidelity curve.

Threshold choice. The thresholds (80% on MOSES, 90% on GuacaMol) were chosen before evaluation, based on reference Novelty levels reported for strong diffusion baselines (DiGress: 95% on MOSES, 99.9% on GuacaMol) and the observation that both H-GVT and GVT† stabilize in the 80–95% Novelty range under continued training. Using a fixed pre-selected threshold (rather than, e.g., maximum-FCD selection) preempts the concern that either model's performance is obtained at a cherry-picked memorization regime.

E.3 Original GVT vs. GVT† vs. H-GVT

Tables 11 and 12 report the three methods side-by-side on MOSES and GuacaMol respectively. Original GVT numbers are taken from [Zheng et al., 2026]; empty cells indicate values not reported by the original authors.

Summary. Under this uniform protocol, H-GVT outperforms GVT† on every metric at matched Novelty, isolating the contribution of hierarchical generation from any advantage obtainable through training-set memorization.

Table 12: **GuacaMol: Original GVT vs. GVT† vs. H-GVT** (official scores; higher is better for KL and FCD). At comparable Novelty ($\sim$92.6%), H-GVT outperforms GVT† on every metric.

Model	Val.↑	Uni.↑	Nov.↑	KL↑	FCD↑
GVT (original) [Zheng et al., 2026]	97.46	99.9	82.39	99.61	85.50
GVT† (ours, Nov. $\geq$ 90%)	92.17	99.9	92.57	96.45	73.24
H-GVT (same rule)	94.70	100.0	92.60	97.94	83.66

F Implementation of Spectral Topology Consistency (SpectralLoss)

To enforce spectral topology consistency between the fine-scale graph and its coarsened counterpart, we compute the eigenvalue spectrum of the normalized graph Laplacian and minimize the discrepancy between the low-frequency spectra.

Normalized Laplacian spectrum. Given a (possibly soft) adjacency matrix $A \in \mathbb{R}^{N\times N}$, we compute the node degree $d_i = \sum_j A_{ij}$ and form the normalized adjacency

$$\tilde{A}_{ij} = d_i^{-\frac{1}{2}} A_{ij} d_j^{-\frac{1}{2}}, \tag{6}$$

where we add numerical safeguards for zero degrees by setting $d_i^{-\frac{1}{2}} = 0$ when $d_i = 0$. The normalized Laplacian is then

$$L = I - \tilde{A}. \tag{7}$$

Since our pooling may produce a nearly-symmetric soft adjacency, we explicitly symmetrize the Laplacian as

$$L \leftarrow \frac{1}{2}(L + L^\top) \tag{8}$$

before eigen-decomposition for numerical stability. We obtain the Laplacian eigenvalues via `torch.linalg.eigvalsh`, which returns them in ascending order.

Padding and active-node cropping. Although batched graph tensors may contain padded nodes for efficient implementation, padded entries are excluded before spectral computation. For each molecule, we first crop the adjacency matrix to the active node set indicated by the node mask and compute the normalized Laplacian only on this active subgraph. Therefore, dummy padded nodes do not appear in the eigen-decomposition and cannot introduce artificial eigenvalues. In the passthrough case ($N \leq K_l$), the active subgraph of the padded level is exactly the original $N \times N$ graph, so the spectral discrepancy for that level is zero.

SpectralLoss definition. Let $\lambda^{(0)}_{1:k}$ be the smallest k eigenvalues of the fine graph Laplacian $L^{(0)}$, and $\lambda^{(l)}_{1:k}$ be those of the coarsened graph Laplacian $L^{(l)}$ at scale l. We define

$$\mathcal{L}^{(l)}_{\text{spec}} = \left\| \lambda^{(0)}_{1:k} - \lambda^{(l)}_{1:k} \right\|_2^2. \tag{9}$$

In our code, the fine graph spectrum is computed on the original active node set. For active pooling levels ($N > K_l$), the coarsened graph has K_l valid cluster nodes, and we compute its spectrum using an all-ones coarse mask. For passthrough levels ($N \leq K_l$), we crop the padded adjacency back to the active $N \times N$ graph before eigen-decomposition; hence this level contributes zero spectral discrepancy.

Choice of k. We set k to focus on low-frequency (global) structure. In practice, we use

$$k = \min\left(k,\, N_{\text{fine}},\, N_{\text{coarse}}\right), \tag{10}$$

and follow our implementation by choosing $k = \lfloor K_l/2 \rfloor$ for scale l, where K_l is the cluster size at that scale.

Remarks. The spectrum is computed on the normalized Laplacian, whose eigenvalues lie in $[0, 2]$, making the loss numerically stable. We use a simple MSE between eigenvalues, which is permutation-invariant and directly encourages the coarse graph to preserve the global connectivity patterns of the fine graph.

F.1 Computational Cost and Solver Choice

$\mathcal{L}_{spec}$ appears only in the Stage-1 objective and is not evaluated during sampling, so the sampling times in Section 4.1 (14.3 s for 10K molecules on QM9) do not include it. For molecular graphs, the dense low-frequency eigendecomposition takes 2–120 μs per molecule, consistent with the 2.8% per-epoch overhead reported in Section 3. Sparse iterative solvers such as `lobpcg` and `eigsh` are designed for large matrices and are three to four orders of magnitude slower than the dense solver at molecular size, so the dense solver is the appropriate choice for the graphs studied in this paper. The cubic cost of the dense solver becomes relevant only for much larger graphs. Table 13 reports the solver times for a graph with $N = 10^4$ nodes. Because $\mathcal{L}_{spec}$ uses only the $k_l = \lfloor K_l/2 \rfloor \leq 16$ smallest eigenvalues, a truncated sparse solver computes the same objective at lower cost.

Solver	Time at $N = 10^4$
Dense (`eigvalsh`)	1.07 s
Sparse top-k (`eigsh`, `lobpcg`)	61–189 ms (6–17$\times$ faster)

Table 13: Eigensolver time for the low-frequency spectrum of a graph with 10^4 nodes.

F.2 Gradient Stability

Losses defined on eigendecompositions can be ill-conditioned when eigenvalues are close, because the derivatives of eigenvectors contain terms proportional to $1/(\lambda_i - \lambda_j)$. $\mathcal{L}_{spec}$ depends only on eigenvalues. For a simple eigenvalue λ_i of $\mathbf{L}$ with unit eigenvector $\mathbf{v}_i$, the derivative with respect to an adjacency entry A_{ab} has the Hellmann–Feynman form

$$\frac{\partial \lambda_i}{\partial A_{ab}} = \mathbf{v}_i^\top \frac{\partial \mathbf{L}}{\partial A_{ab}} \mathbf{v}_i, \tag{11}$$

which contains no $1/(\lambda_i - \lambda_j)$ term, and the loss does not backpropagate through eigenvectors. We verified this numerically by computing $\partial \mathcal{L}_{spec}/\partial \mathbf{A}$ with the dense solver and with the sparse solver (`lobpcg`) on 400 ZINC250k molecules. The cosine similarity between the two gradients is 1.00000, the relative L_2 error is about 5×10^{-6}, and the gradient norms are identical. Replacing the dense solver with a sparse top-k solver therefore does not measurably change the gradient.

G Structural Content of Coarse Codes

G.1 Mechanism

This appendix expands the brief argument in Section 3 on why coarse codes $\mathbf{Z}_q^{(l)}$, which never directly enter the decoder, can still provide useful structural context for Stage 2 generation. We first emphasize the scope of the claim: H-GVT does not require, and we do not claim, that a particular slot corresponds to a fixed chemical motif across molecules. Instead, the RCM-ordered input and fixed GNN output channels define a reproducible slot-indexed layout, while the pooling and spectral losses encourage each slot to encode a localized coarse region of the molecular graph.

Three mechanisms then shape the content carried by these slots. First, all scales share a single codebook $\mathcal{C}$, and the dominant gradient on $\mathcal{C}$ comes from $\mathcal{L}_{recon}$, so its entries are tied to atom-level chemistry; quantizing any coarse embedding against this codebook maps it into the same discrete vocabulary used by fine-scale generation. Second, each coarse embedding $\mathbf{z}_j^{(l)} = \sum_i S_{ij}^{(l)} \mathbf{z}_i^{(0)}$ is a soft aggregation of fine-scale vectors, with $\mathcal{L}_{mc}^{(l)}$ and $\mathcal{L}_{ortho}^{(l)}$ encouraging well-connected and non-overlapping pooled regions. Third, $\mathcal{L}_{spec}$ constrains the coarsened graph to preserve the low-frequency Laplacian spectrum of the original, anchoring the coarse representation to global topology. The ablations in Section 4.3 are consistent with this account: randomizing the coarse-to-fine order degrades NSPDK by 29$\times$, indicating that the slot-indexed order carries useful structural information, while removing $\mathcal{L}_{spec}$ or adaptive pooling separately costs 3$\times$ and 6$\times$ on NSPDK. Appendix G.3 provides qualitative examples where coarse clusters align with rings and functional groups.

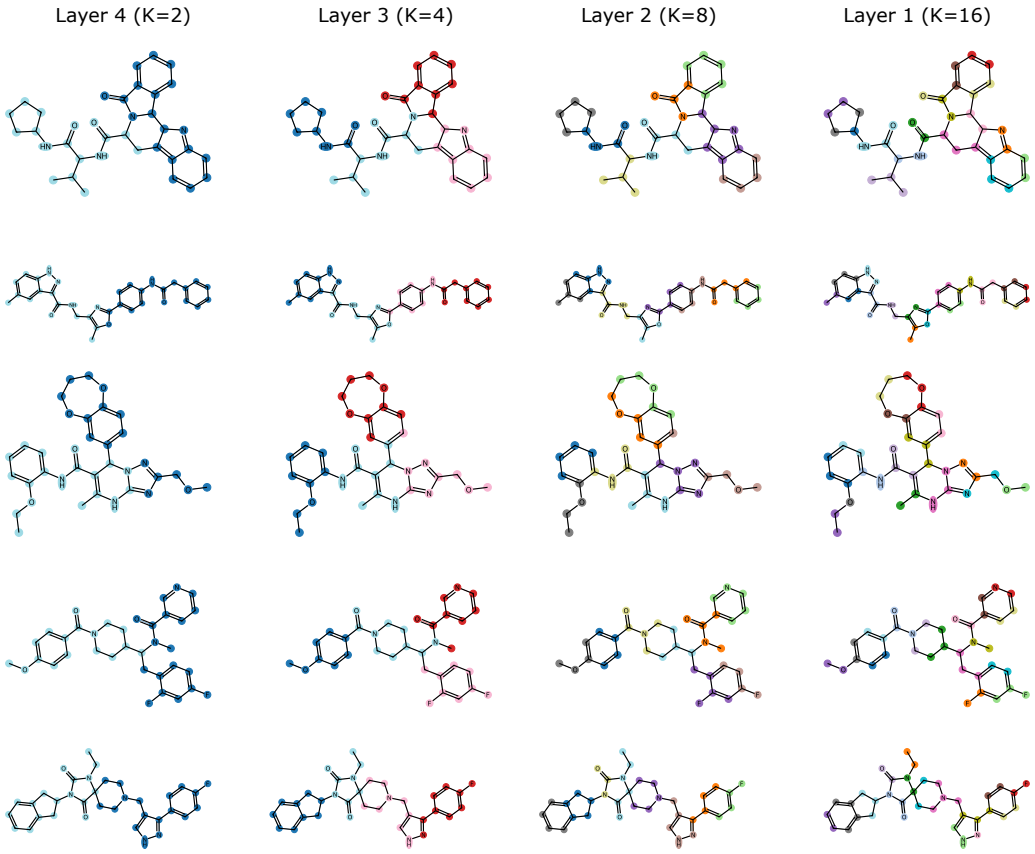

Figure 4: Visualization of learned hierarchical coarsening for representative generated molecules.

G.2 Coarse-to-Fine Refinement Trajectory

To illustrate the behavior of our coarse-to-fine generation approach, we provide additional visualizations for five representative molecules in Figure 4. Each row depicts a hierarchical trajectory where the model first constructs a macroscopic skeleton (coarsest scale) and progressively resolves it into localized functional groups and individual atoms.

The color-coding in these visualizations shows that the hierarchical clustering is not merely a mathematical partitioning but aligns well with chemical intuition. For instance, the model frequently groups atoms belonging to the same aromatic ring or side chain into single clusters at coarser scales. The coarse-to-fine paradigm constrains the search space for atom placement within the global context established at higher levels. This hierarchical dependency helps the generated molecules maintain structural integrity and avoids the fragmented or disjointed graphs that often arise in non-hierarchical, atom-by-atom generation methods.

G.3 Single-Molecule Case Study

Figure 5 zooms in on a single ZINC250k molecule to expose the internal mechanics of the hierarchy, complementing the multi-molecule overview in Figure 4. At the coarsest level (K=2), the model captures the fundamental topological partition of the molecule, separating its two largest substructural regions. At finer scales, these coarse super-nodes are expanded into distinct functional groups and ring systems, with the assignment pattern remaining chemically coherent across levels. The soft assignment matrices $\mathbf{S}$ exhibit high-confidence, sparse assignments: each atom concentrates its mass on a small number of clusters, and the cluster rows are near-orthogonal. This confirms that the Orthogonality Loss imposed during Stage-1 training induces non-overlapping, chemically meaningful clusters rather than diffuse or overlapping assignments, validating the interpretability claims made in

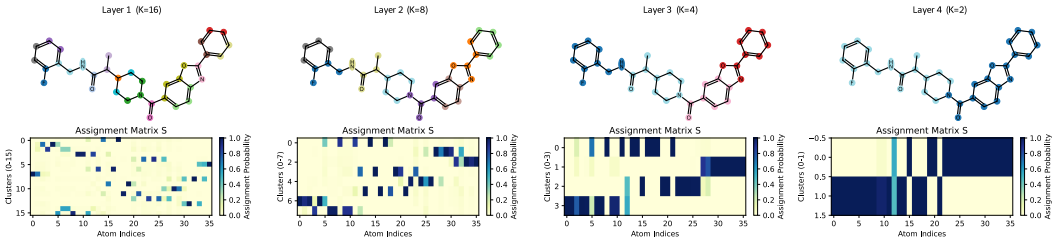

Figure 5: Visualization of learned hierarchical coarsening. Top row: molecular graph at four scales (coarse $K{=}2$ to fine $K{=}16$), color-coded by cluster assignment. Bottom row: soft assignment matrices $\mathbf{S}$, mapping atoms (x-axis) to clusters (y-axis). The adaptive pooling identifies structurally coherent super-nodes (rings, functional groups) that provide coarse structural context for fine-grained generation.

Section 4.4 and providing the per-molecule visual counterpart to the aggregate evidence presented there.

G.4 Quantitative Chemical Alignment of Learned Clusters

The qualitative visualizations in the previous section suggest that learned clusters often correspond to chemically coherent substructures. Here we provide a quantitative validation of this observation. For each molecule, we extract individual rings, aromatic rings, ring systems, and common functional groups using RDKit substructure matching. Given the soft assignment matrix $\mathbf{S}^{(l)} \in \mathbb{R}^{N \times K_l}$ at hierarchy level l, we measure how well a chemical substructure G is captured by the best learned cluster.

For a cluster j, we define the soft intersection between G and cluster j as

$$I_{G,j}^{(l)} = \sum_{i \in G} S_{ij}^{(l)}. \tag{12}$$

The corresponding soft precision, recall, and F1 score are

$$\mathrm{Prec}_{G,j}^{(l)} = \frac{I_{G,j}^{(l)}}{\sum_i S_{ij}^{(l)}}, \qquad \mathrm{Rec}_{G,j}^{(l)} = \frac{I_{G,j}^{(l)}}{|G|}, \tag{13}$$

$$\mathrm{F1}_{G,j}^{(l)} = \frac{2 \cdot \mathrm{Prec}_{G,j}^{(l)} \cdot \mathrm{Rec}_{G,j}^{(l)}}{\mathrm{Prec}_{G,j}^{(l)} + \mathrm{Rec}_{G,j}^{(l)}}. \tag{14}$$

We then define the chemical alignment score of G as the best soft F1 over all clusters:

$$\mathrm{AlignF1}_{G}^{(l)} = \max_j \mathrm{F1}_{G,j}^{(l)}. \tag{15}$$

This metric is high only when a chemical substructure is mostly contained within one learned cluster and that cluster does not include many unrelated atoms.

We compare H-GVT with two non-chemical baselines. The first is a random soft-assignment baseline with the same number of clusters K_l. The second is a stronger within-molecule permutation baseline, which shuffles atom identities while preserving the learned cluster mass distribution. Single-atom groups such as halides are excluded from the main panels because their alignment is trivially high under hard assignments.

Overall, the quantitative results support the interpretation that the learned hierarchy captures chemically coherent substructures. Importantly, this does not imply that a fixed cluster slot corresponds to the same chemical motif across all molecules. Rather, the learned clusters form molecule-specific coarse regions that are statistically aligned with rings and functional groups, providing chemically meaningful structural context for coarse-to-fine autoregressive generation.

Comparison with contiguous RCM segments. We add a third baseline that partitions the RCM-ordered atom sequence into K_l contiguous segments. Because the RCM order places graph-adjacent

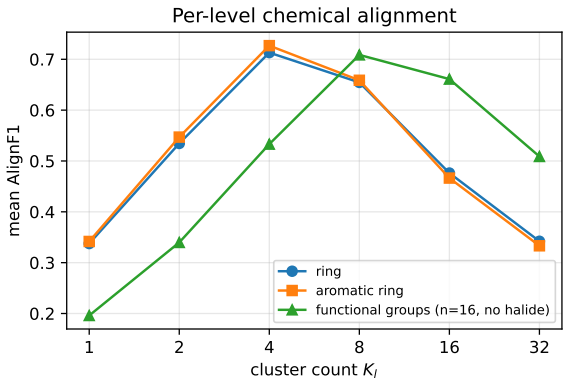

Figure 6: **Per-level chemical alignment of learned clusters.** We report the mean AlignF1 between learned clusters and RDKit-detected chemical substructures across hierarchy levels. Functional groups exclude single-atom halides. The learned hierarchy shows scale-dependent chemical organization: intermediate levels align strongly with local functional groups, whereas coarser levels better capture ring and aromatic-ring structures.

Substructure	$\mathbf{K_1}$	**#Inst.**	**H-GVT**	**Random**	**Perm.**
Carboxyl	8	485	**0.827**	0.206	0.376
Sulfonyl	8	747	**0.802**	0.206	0.377
Sulfonamide	8	561	**0.784**	0.219	0.378
Ester	8	864	**0.712**	0.217	0.373
Amide	8	3968	**0.704**	0.203	0.372
Ether	8	4188	**0.703**	0.205	0.374
Aromatic ring	4	12445	**0.727**	0.304	0.394
Ring	4	17312	**0.713**	0.301	0.392
Ring system	4	3078	**0.698**	0.370	0.439

Table 14: **Representative chemical-alignment scores.** We report mean AlignF1 for selected functional groups and ring structures at their most informative hierarchy levels. At $K_l = 8$, learned clusters align strongly with local functional groups such as carboxyl, sulfonyl, ester, amide, and ether groups. At $K_l = 4$, learned clusters better capture larger ring-level structures. Both trends are substantially above the random soft-assignment baseline and the stronger within-molecule permutation baseline.

atoms close together, this baseline tests whether the alignment is a consequence of the atom order and not of learned pooling. Table 15 reports the mean AlignF1 over the 16 functional groups at $K_l = 8$ on 5,000 GuacaMol molecules. The learned clusters obtain 0.726, compared with 0.616 for contiguous RCM segments, 0.382 for within-molecule permutation, and 0.206 for random soft assignments. Each of the 16 functional groups obtains a higher AlignF1 with the learned clusters than with the RCM segments at the resolution that matches its size, with differences between 0.03 and 0.23. The best-matching resolution depends on the size of the substructure. Compact groups of two or three atoms reach their highest alignment at $K_l = 16$, mid-sized groups at $K_l = 8$, and ring systems at $K_l = 4$.

Method	Mean AlignF1
H-GVT learned clusters	**0.726**
RCM-contiguous segments	0.616
Within-molecule permutation	0.382
Random soft assignment	0.206

Table 15: Mean AlignF1 over 16 functional groups at $K_l = 8$ (5,000 GuacaMol molecules).

G.5 Linear Probes on Frozen Codes

We freeze the ZINC250k Stage-1 model and extract the coarse codes ($l \geq 1$) and the fine codes ($l = 0$) of 60K ZINC250k molecules. For each of the six global descriptors in Table 5, we fit a linear probe on each code set and report the held-out R^2, or the classification accuracy for the 40 most frequent Bemis–Murcko scaffolds. The shuffled control, which removes the correspondence between codes and targets, obtains $R^2 = 0.00$ on all regression targets and an accuracy equal to the majority-class rate (0.34) on the scaffold target. Concatenating the coarse and fine codes gives the highest accuracy, with $R^2 = 0.85$ on ring count compared with 0.72 for the coarse codes and 0.76 for the fine codes alone.

Table 16 reports the code usage of each pooled level of the ZINC250k tokenizer. The $K_l = 16$ level uses the most diverse code distribution (346 distinct codes, 6.93 bits of entropy). The coarser levels use smaller vocabularies (87 to 172 distinct codes, 2.26 to 3.14 bits), none of which collapses to a single code.

Level	K_l	Distinct codes used	Code entropy (bits)
$l = 5$	1	87	2.65
$l = 4$	2	172	3.14
$l = 3$	4	115	2.37
$l = 2$	8	131	2.26
$l = 1$	16	346	6.93

Table 16: Code usage of the pooled levels of the ZINC250k tokenizer.

G.6 Causal Intervention on the Coarse Prefix

We take 300 coarse prefixes, each consisting of the coarse codes ($l \geq 1$) of a real molecule from GuacaMol [Brown et al., 2019]. For each prefix, we keep the prefix fixed and sample the fine level 16 times with the Stage-2 prior at a fine-scale temperature of 1.3, which gives 4,800 completions. Each completion is decoded with the Stage-1 decoder, and its ring count is computed. Of the completions, 46% are unique molecules. Of the completions that share a prefix, 90.7% have the same ring count. The intraclass correlation coefficient of ring count, computed with prefixes as groups, is 0.88, so 88% of the ring-count variance lies between prefixes and 12% within prefixes. The correlation between the ring count of a completion and that of the source molecule of its prefix is 0.97. A fixed coarse prefix therefore determines most of the ring topology of the completions, while the fine level produces distinct molecules. This result is consistent with the small effect of the fine-scale temperature on global metrics (Appendix J).

H Standard vs. Scale-Causal Masking

A natural alternative to the standard causal mask used in Section 3.2 is a scale-causal mask, in which tokens attend bidirectionally within their own hierarchy level and only causally across levels. We investigated this variant and found that it substantially degrades performance: on ZINC250k, the scale-causal mask reduces validity to 87.99% (vs. 99.86%), raises FCD to 1.81 (vs. 1.15), and worsens NSPDK by roughly an order of magnitude (Table 4). The reason is that the coarse-to-fine hierarchy in H-GVT is encoded in the data representation (token ordering plus level embeddings), not in the attention pattern. The standard causal mask already ensures that every fine-scale token conditions on all coarse-scale predecessors, which is exactly the dependency structure required. Imposing additional architectural constraints through a specialized mask adds optimization difficulty without providing any modeling benefit that the ordering does not already supply.

I Stage-2 Transformer Model Scaling and Generation Quality

In this section, we investigate how the capacity of the Stage-2 Transformer affects the quality of the generated molecular distributions. We vary the embedding dimension (d_{model}) and the number of layers (L) while keeping other training hyperparameters constant (e.g., training for 400 epochs). Table 17 summarizes the configurations evaluated.

#Layers	Embed Dim	Heads	Params (M)	Val.↑	Uni.↑	Nov.↑	KL↑	FCD↑
12	512	16	40.0	94.6	99.99	95.96	97.87	81.31
12	560	16	47.7	94.9	99.97	93.65	98.02	82.66
12	640	16	62.0	95.24	99.96	87.91	98.57	83.27
12	768	16	88.4	97.01	99.89	61.03	99.13	85.41
12	800	16	95.9	97.7	99.86	55.30	99.43	85.84

Table 17: Performance comparison of GuacaMol dataset across different Stage-2 Transformer architectures. All models were trained for 400 epochs. Params denotes the number of trainable parameters in millions.

Table 17 reports the GuacaMol distribution-learning results under different Stage-2 Transformer capacities. Overall, increasing the embedding dimension consistently improves distribution-matching metrics such as KL and FCD, indicating a stronger ability to model the target molecular distribution as model capacity grows. Specifically, KL improves from 97.87 to 99.43 and FCD improves from 81.31 to 85.84 when scaling d_{model} from 512 to 800.

Interestingly, validity also benefits from scaling, rising from 94.6% to 97.7%. In contrast, novelty exhibits a clear degradation at larger model sizes: while small-to-medium models maintain high novelty (e.g., 95.96 at d_{model}=512), novelty drops sharply once $d_{model} \geq 768$ (61.03 at 768 and 55.30 at 800). This suggests that overly large Transformer decoders tend to overfit the training distribution and generate samples with reduced novelty, even though they achieve better distribution-level matching scores. Uniqueness remains nearly saturated (all ≈ 99.9) across all configurations, implying that scaling does not meaningfully affect diversity under this metric.

Based on these results, there exists a practical trade-off between distribution fidelity (KL/FCD) and novelty when scaling the Stage-2 Transformer. We therefore adopt a moderate-to-large configuration (e.g., L=12, d_{model}=640) as a balanced choice, which achieves strong KL/FCD (98.57/83.27) while preserving substantially higher novelty (87.91) compared to the largest models.

J Impact of Scale-Specific Sampling Temperatures

J.1 Motivation and Experimental Design

To evaluate the influence of sampling stochasticity on hierarchical generation, we conduct a grid search on a high-capacity variant of the Stage-2 Transformer (exceeding 100M parameters). This model was chosen because it shows localized overfitting under standard settings ($T = 1.0$), with a lower novelty score and a tendency to reproduce training scaffolds. Stress-testing this model lets us investigate whether scale-specific temperature tuning can reduce memorization and recover structural diversity.

We define two parameters. The coarse-scale temperature T_1 governs the joint probability of hierarchical discrete codes at macroscopic levels ($l \in \{L, \ldots, 1\}$), while the fine-scale temperature T_2 controls the sampling of the final atom-level resolution ($l = 0$). All evaluations were performed on the MOSES dataset using Top-$k = 50$ sampling with 11,000 generated molecules per configuration.

J.2 Observations and Discussion

The cross-analysis heatmaps in Figure 7 provide several useful insights into the hierarchical generative flow.

Mitigating memorization via T_1. A key finding is that structural novelty is predominantly driven by T_1. For this overfitted model, $T_1 = 0.5$ results in a Novelty score of only $\approx 2\%$, indicating severe memorization of training skeletons. Increasing T_1 to 1.5 recovers the Novelty score to $\approx 39\%$. This shows that the "overfitting" is primarily localized in the coarse-scale topological transitions, and increasing T_1 allows the model to escape these high-probability memorized paths.

Optimal fidelity–diversity trade-off. The Fréchet ChemNet Distance (FCD) shows a distinct sensitivity to T_1. The optimal distribution matching (FCD $= 0.119$) occurs at $T_1 = 1.0, T_2 = 0.5$. The model maintains high structural validity ($> 98\%$) even at the highest temperatures, confirming

that the Stage-1 VQ-VAE decoder provides a stable chemical manifold that constrains the output regardless of sampling noise.

The stability of fine-scale refinement. Compared to T_1, the impact of T_2 on global metrics (e.g., Scaffold similarity and SNN) is relatively marginal. This supports our hypothesis that once the global topological "map" is established at the coarse scales, the fine-scale completion is a highly constrained process where local valency rules dominate over global stochasticity.

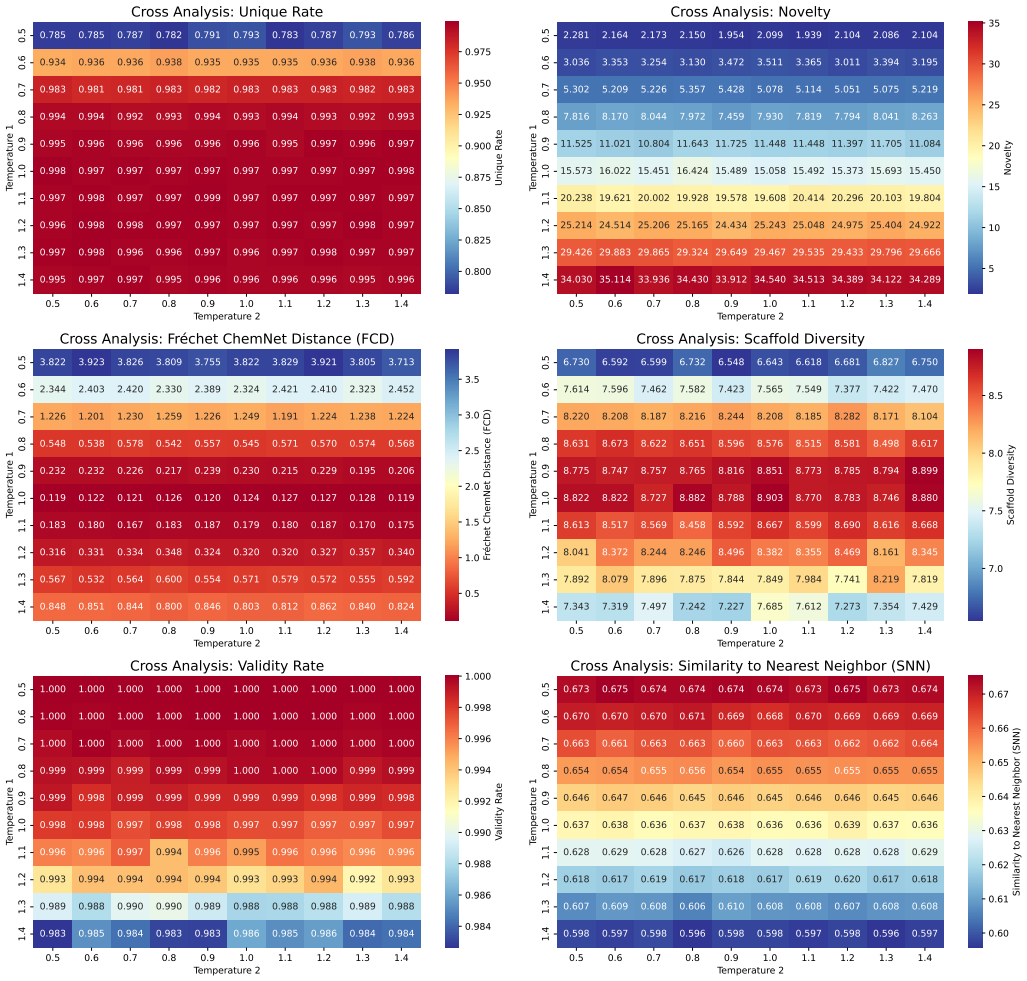

Figure 7: Cross-analysis of sampling temperatures T_1 (coarse) and T_2 (fine) on the MOSES dataset. Even for a high-capacity model prone to overfitting, increasing T_1 effectively recovers structural novelty while maintaining competitive distribution matching (FCD).

K Property Distribution Analysis

Beyond aggregate metrics such as FCD and NSPDK, we evaluate how well generated molecules reproduce the marginal distributions of individual physicochemical properties. We conduct this analysis on GuacaMol, which contains ∼1.6M drug-like molecules from ChEMBL [Gaulton et al., 2012] and thus represents the most realistic, large-scale benchmark in our evaluation. Figure 8 compares kernel density estimates (KDE) across 11 properties, grouped into two tiers. The first tier covers six representative pharmacochemical descriptors: lipophilicity (MolLogP), molecular weight (MolWt), topological polar surface area (TPSA), drug-likeness (QED) [Bickerton et al., 2012], synthetic accessibility (SA Score) [Ertl and Schuffenhauer, 2009], and molecular complexity (BertzCT) [Bertz, 1981]. The second tier covers five additional structural counts: hydrogen bond

acceptors (NumHAcceptors) and donors (NumHDonors), rotatable bonds (NumRotatableBonds), aliphatic rings (NumAliphaticRings), and aromatic rings (NumAromaticRings).

H-GVT fidelity. H-GVT closely matches the training distribution across all 11 properties, with KL divergences uniformly low (ranging from 0.0024 for MolLogP to 0.0166 for BertzCT; 8 of 11 properties fall below 0.01). These values substantially exceed the aggregate-metric improvements reported in the main text and confirm that H-GVT's gains are not limited to graph-topology statistics but extend to fine-grained molecular property distributions relevant for downstream drug-discovery workflows.

DiGress deviations. DiGress shows systematic deviations on several pharmacologically important properties. On SA Score, DiGress produces a flattened, right-shifted distribution with substantial density at SA$>$4, indicating a tendency to generate molecules that are harder to synthesize, whereas H-GVT reproduces the sharp training-set mode near SA$\approx$2.5 with high fidelity. On QED, DiGress under-represents the high drug-likeness tail (QED$>$0.7) and over-represents low-QED molecules, while H-GVT maintains close agreement throughout the range. On BertzCT, DiGress generates a broader right tail, drifting toward higher topological complexity than the training set. For the integer-valued properties (NumHAcceptors, NumHDonors, NumAliphaticRings, NumAromaticRings), DiGress over-smooths the discrete peaks, whereas H-GVT accurately reproduces their multi-modal structure.

GVT† deviations. GVT†, the Novelty-thresholded reproduced baseline introduced in Appendix E, tracks the training distribution on continuous descriptors such as MolLogP, TPSA, and QED but displays its own distinctive failure modes on scale- and count-sensitive properties. On MolWt, GVT† shifts the mode noticeably to the left and inflates the low-molecular-weight tail, indicating a bias toward smaller molecules. On SA Score, it flattens the sharp training-set mode in a manner qualitatively similar to DiGress, losing density near SA$\approx$2.5 and adding mass in the 3–5 range. On BertzCT, GVT† shifts left with a heavier low-complexity tail, the opposite bias of DiGress's right-shifted tail. On the integer-valued ring counts (NumAliphaticRings, NumAromaticRings), GVT† systematically under-represents the dominant peaks. Taken together, these patterns suggest that flat autoregressive decoding over a single-level vocabulary, even when the Stage-2 Transformer capacity is matched to H-GVT and the checkpoint is selected under the same Novelty-thresholded protocol, struggles to preserve the size and ring-count statistics of drug-like molecules, a limitation that the hierarchical coarse-to-fine factorization in H-GVT addresses directly.

K.1 Structural Alerts and Scaffold Novelty

We evaluate two medicinal-chemistry criteria on GuacaMol. The pass rate is the fraction of valid molecules with zero PAINS or Brenk structural alerts, computed separately for each alert set. Scaffold novelty is the fraction of generated Bemis–Murcko scaffolds that do not occur in the scaffold set of the full training set.

Source	PAINS pass	Brenk pass	Scaffold novelty
H-GVT	95.2%	61.0%	61.5%
GVT†	94.3%	53.5%	57.7%
DiGress	95.6%	57.9%	78.9%
Training set (reference)	94.0%	57.0%	–

Table 18: Structural-alert pass rates and scaffold novelty on GuacaMol.

Table 18 shows that H-GVT does not increase the incidence of either alert class relative to the training set (PAINS pass rate of 95.2% vs. 94.0%, Brenk pass rate of 61.0% vs. 57.0%). The scaffold novelty of H-GVT is higher than that of GVT† (61.5% vs. 57.7%) and lower than that of DiGress (78.9%). This ordering follows the overall Novelty of the models (99.9% for DiGress and 92.6% for H-GVT in Table 3), and DiGress obtains a lower FCD score (68.0 vs. 83.66), which is the fidelity–diversity trade-off quantified in Appendix E. A retrosynthesis-based evaluation requires an external retrosynthesis planner that we have not validated for this benchmark. We report the SA score (Figure 8) as the standard proxy and leave retrosynthetic evaluation to future work.

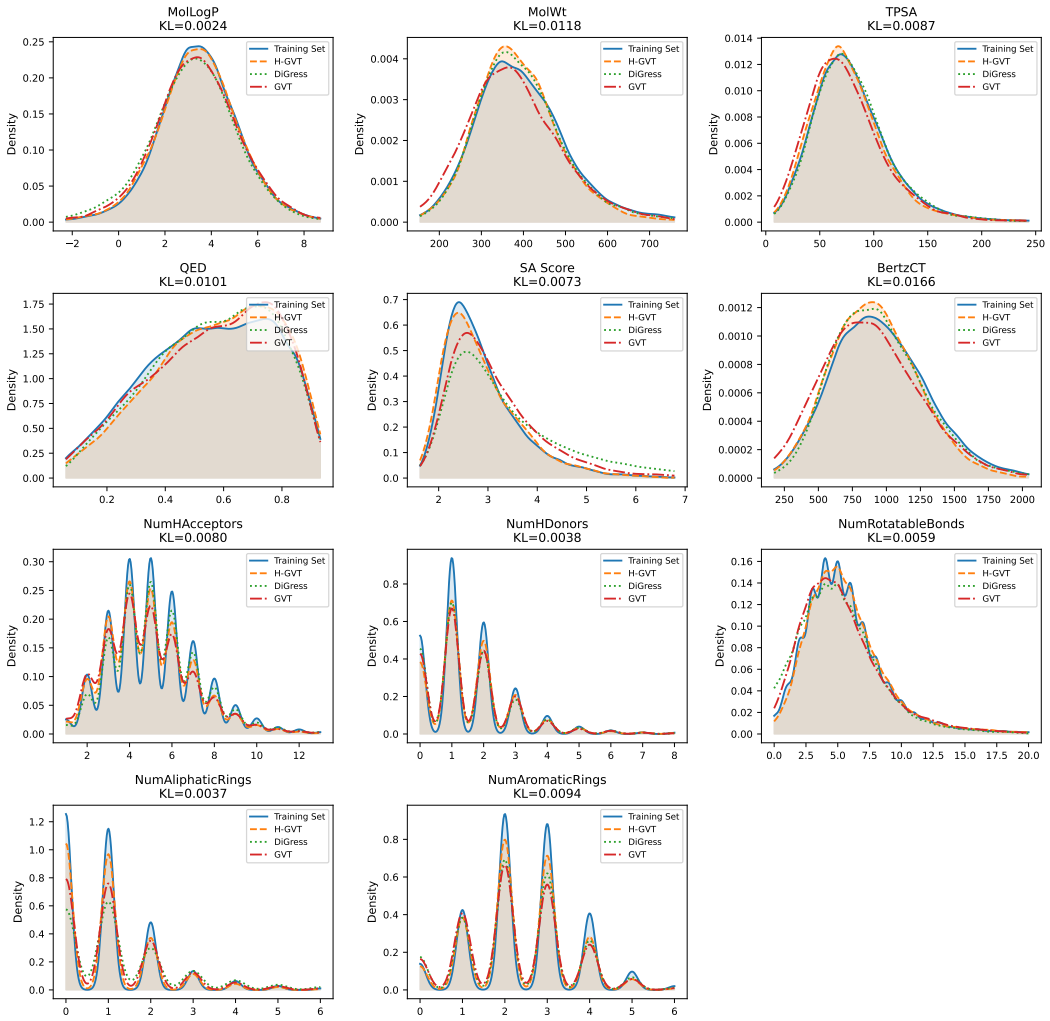

Figure 8: **Physicochemical property distributions (11 properties) of generated molecules vs. the GuacaMol training set.** KDE plots comparing H-GVT (orange, dashed), DiGress (green, dotted), and GVT† (red, dash-dotted) against the training distribution (blue, solid). KL divergence from the training set is shown for H-GVT in each subplot title. H-GVT consistently tracks the training distribution across all 11 properties; DiGress and GVT† exhibit distinct failure modes, most visibly on SA Score, MolWt, BertzCT, and the integer-valued ring counts.

L Property-Conditioned Generation

Formulation. For a molecule with properties $(\mathrm{QED}, \log P)$, we bin each property into deciles $b_q, b_p \in \{1, \ldots, 10\}$ and learn two embedding tables $\mathbf{E}_q, \mathbf{E}_p \in \mathbb{R}^{10\times d}$. The conditioning vector

$$\mathbf{c} = \mathbf{E}_q[b_q] + \mathbf{E}_p[b_p] \in \mathbb{R}^d \tag{16}$$

is added at every position, including the start-of-sequence position $i = 0$, so that Eq. (5) becomes

$$h_i = \mathrm{TokEmb}(t_i) + \mathrm{PosEmb}(i) + \mathrm{LevelEmb}(\ell_i) + \mathbf{c}, \qquad i = 0, \ldots, |\tau|, \tag{17}$$

and the factorization becomes $p(\tau \mid \mathbf{c}) = \prod_t p(\tau_t \mid \tau_{<t}, \mathbf{c})$. The two embedding tables are the only change. The architecture, tokenizer, codebook, and causal mask are unchanged.

Setup. Stage 1 is the frozen ZINC250k tokenizer, and only the Stage-2 prior is trained on ZINC250k. This experiment is a feasibility study with a deliberately small prior (8 layers, $d_{model} = 256$, 7.4M

parameters, 100 epochs), whose training takes 1 h 50 min on a single RTX 5090 GPU. Each property is swept over its 10 decile conditions with the other property held at a middle decile, and 800 molecules are sampled per condition.

Target	Pearson r	MAE (all bins)	MAE (bins 2–9)	Realized, bin 1 $\rightarrow$ 10	Validity	Uniqueness
QED	0.966	0.041	0.017	0.564 $\rightarrow$ 0.872	88.5–98.6%	100%
$\log P$	0.988	0.397	0.077	0.931 $\rightarrow$ 4.410	95.5–98.1%	100%

Table 19: Property-conditioned generation on ZINC250k with a frozen Stage-1 tokenizer. The MAE is the tracking error between the target and realized property values over the decile conditions.

Results. Table 19 shows that the realized properties increase monotonically with the conditioning bin and depend almost linearly on it across the interior bins. The MAE over all bins is dominated by the two boundary bins, whose realized values lie closer to the mean than their targets (QED bin 1 with a target of 0.341 and a realized value of 0.564, $\log P$ bin 10 with a target of 5.787 and a realized value of 4.410). These tail bins have center values that lie far from most molecules in the bin, and the corresponding extreme joint (QED, $\log P$) cells are the least populated cells of the training set. Validity is 95–99% for the central conditions and decreases to 88.5% only for the most extreme QED condition, and uniqueness is 100% for all conditions. Each new conditioning signal requires only a Stage-2 run, because the tokenizer is trained once.

M Effect of Molecule Size

Benchmark	Mean heavy atoms	Levels	FCD of H-GVT vs. best flat AR baseline
QM9	8.80	3	comparable (0.87 vs. 0.86–0.87)
MOSES	21.64	5	$4.8\times$ lower (0.19 vs. 0.92)
ZINC250k	23.11	5	lowest overall (1.147 vs. 1.16)
GuacaMol	27.94 (max. 88)	6	+14.2% (83.66 vs. 73.24)

Table 20: FCD of H-GVT relative to the best flat AR baseline for benchmarks of increasing molecule size. FCD is the raw distance (lower is better), except on GuacaMol, where the official FCD score (higher is better) is reported. The baseline on MOSES and GuacaMol is GVT†.

QM9 molecules contain at most 9 heavy atoms, and H-GVT uses three levels on QM9 ($K_l \in \{4, 2, 1\}$), so a single coarse level frequently summarizes the whole molecule. The three latent AR models obtain FCD values between 0.86 and 0.87 on QM9, which suggests a limitation shared by this model family for small molecules and not a degradation specific to H-GVT. We expect FCD on molecules with at most 9 heavy atoms to depend mainly on atom composition and small-substructure statistics, which flat AR models already capture. On QM9, H-GVT still obtains the lowest NSPDK among all compared methods (2×10^{-4}), and NSPDK measures the match of graph-topology distributions. Table 20 shows that the FCD advantage of H-GVT appears on the three benchmarks with larger molecules and five or six levels.

A fixed per-dataset schedule is over-specified for the 9-atom molecules of QM9 and may be insufficient for the largest GuacaMol molecules with 88 heavy atoms. Per-molecule adaptive choices of L and K_l are compatible with the coarse-to-fine factorization but require an extension of the fixed coarse-token layout, for example through level masks or level-specific termination tokens. We leave this extension to future work.

N Generated Molecular Samples

In this section, we provide qualitative visualizations of molecular graphs generated by H-GVT on all four benchmarks. All molecules shown are randomly selected from the model outputs produced under the same experimental settings as in the main results (i.e., identical preprocessing, sampling configuration, and evaluation protocol). These galleries serve as a complementary sanity check, illustrating the diversity and chemical plausibility of the generated structures.

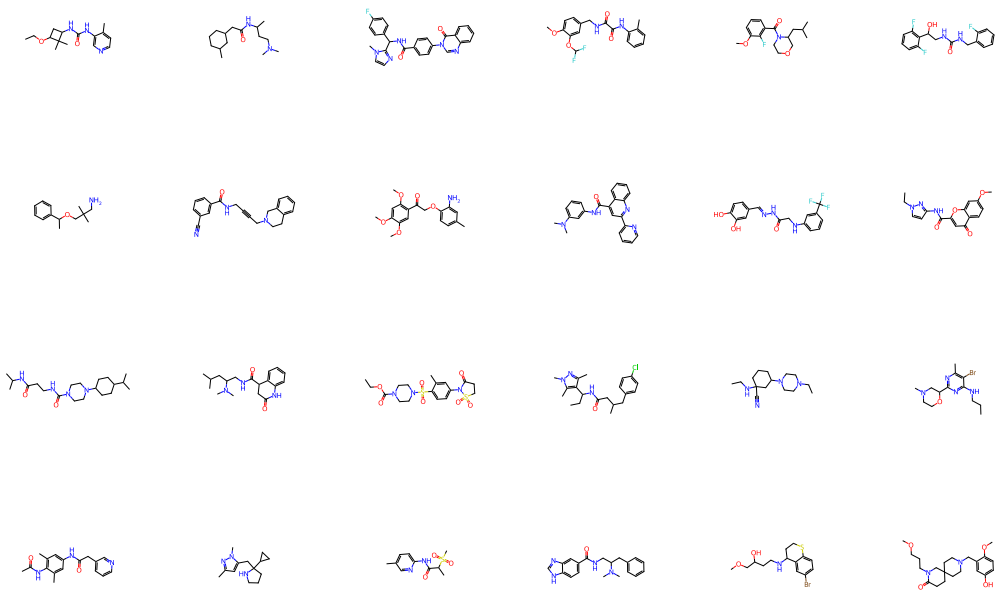

Figure 9: Randomly selected molecules generated by H-GVT on ZINC250k.

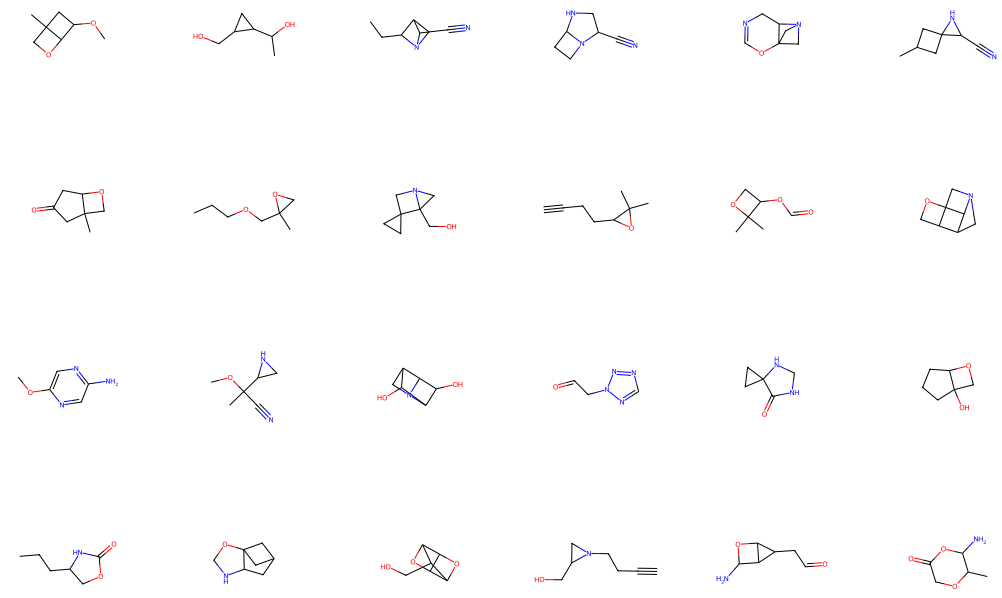

Figure 10: Randomly selected molecules generated by H-GVT on QM9.

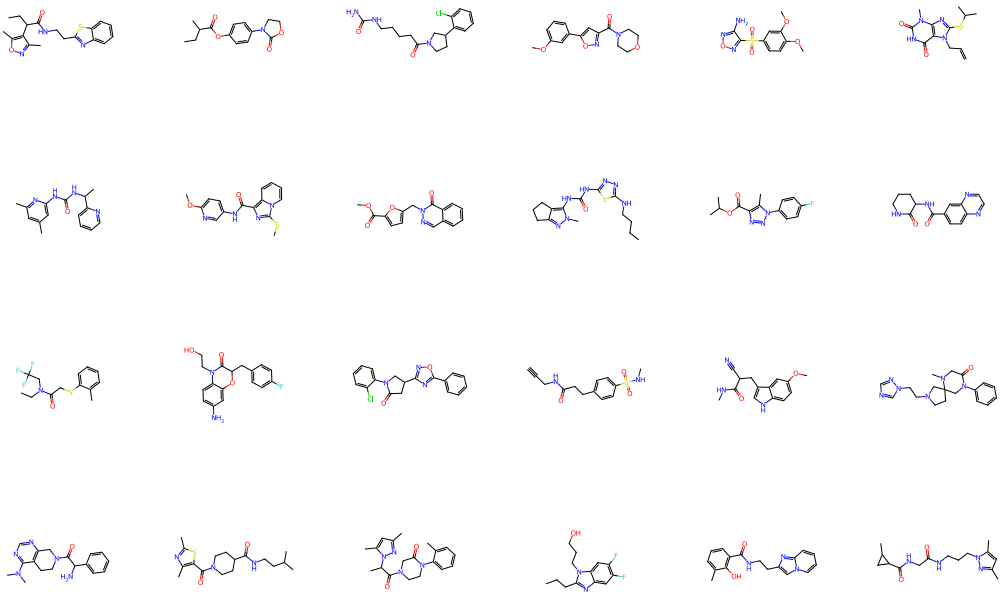

Figure 11: Randomly selected molecules generated by H-GVT on MOSES.

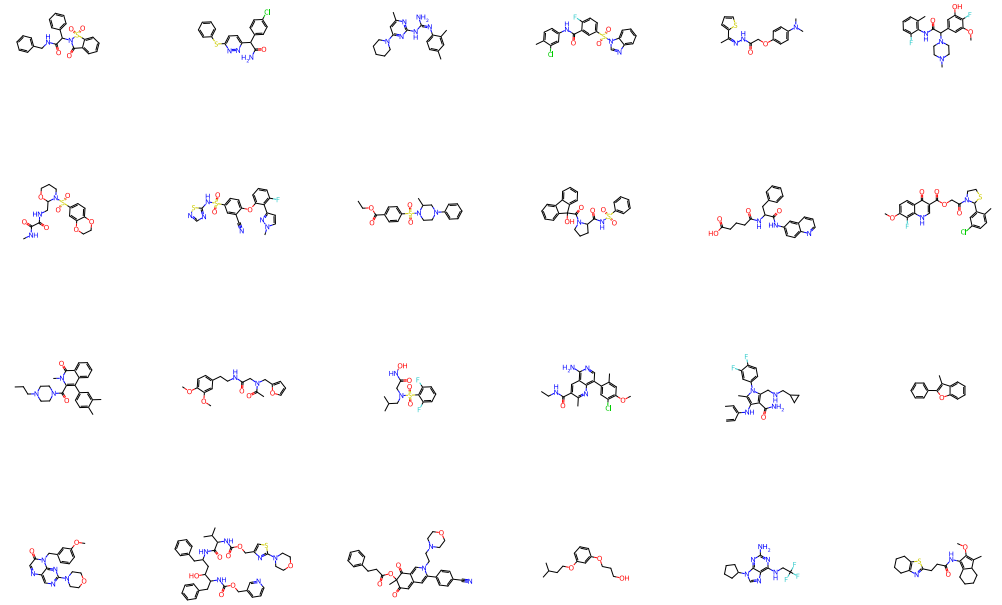

Figure 12: Randomly selected molecules generated by H-GVT on GuacaMol.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction are supported by the experiments in Section 4.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 discusses the limitations: the non-unique spectral prior, the $O(N^3)$ eigendecomposition, the restriction to 2D graphs, and the training cost.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper contains no formal theoretical results.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Appendices A and C give the data processing, evaluation protocol, and hyperparameters, and Algorithms 1–3 give the pseudocode.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets are public benchmarks, and code, configurations, and checkpoints are available at `https://github.com/zzccppp/H-GVT`.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Data splits are given in Section 4, and all hyperparameters and sampling settings in Appendix C.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report means over three sampling runs (Appendix A.4) but no error bars, because most baseline numbers are single values taken from prior work.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: Appendix A.3 gives the hardware, Appendix A.5 the training time, and Section 4.1 the sampling time.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses only public benchmark datasets and involves no human subjects or private data.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This is foundational methods research evaluated on standard benchmarks, with no direct path to negative societal impact.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released models generate small molecular graphs and do not pose a high risk of misuse.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets, software libraries, and baselines are cited to their original publications.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code, configurations, and checkpoints are documented in the repository README at `https://github.com/zzccppp/H-GVT`.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing or human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper involves no human subjects.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not a component of the method.