A Unified Diffusion–Autoregressive Framework for Permutation-Agnostic Graph Synthesis

Usman Ahmad Usmani
Xiamen University Malaysia
Bandar Sunsuria, 43900 Sepang, Malaysia
ahmadusmani321@gmail.com

Arunava Roy
The University of Memphis,
Memphis, TN-38152, USA
royism.arunava@gmail.com

Ari Happonen*
LUT, Finland
53850 Lappeenranta Finland
ari.happonen@lut.fi

***Abstract*—Graph generative modeling faces a long-standing dichotomy: autoregressive (AR) methods offer high expressivity but suffer from node-order sensitivity, while diffusion-based models ensure permutation invariance yet lose directional coherence. We propose PARDIFF, a novel Progressive AutoRegressive Diffusion framework that unifies these paradigms through block-wise, order-agnostic synthesis driven by learned structural decomposition. Unlike prior heuristic-based or strictly sequential approaches, PARDIFF simultaneously predicts block sizes, ranks node positions, and applies an equivariant diffusion process within each block, aligning the strengths of AR guidance with the robustness of score-based sampling. This formulation reframes graph generation as probabilistic inference over learned topological partitions, enabling scalable and semantically faithful synthesis across molecular and non-molecular domains, without relying on auxiliary node features or templates. Extensive experiments on molecular, and social graph benchmarks demonstrate that PARDIFF achieves state-of-the-art performance in both fidelity and diversity metrics. Moreover, its modular and latency-aware design supports real-time deployment in high-impact settings such as drug–drug interaction prediction. We position PARDIFF as a new foundation for permutation-robust, directionally controlled graph generation. In order to promote reproducibility, our codes are open-sourced at: GitHub.**

***Index Terms*—AR diffusion, permutation-invariant graph generation, molecular graph synthesis, block-wise generation.**

I. Introduction

Graphs encode discrete relational structure in domains ranging from social interaction networks and recommender systems to biochemical pathways and cyber–physical infrastructures [19]. As learning paradigms move toward foundation-level generative models, graph generation has become a core primitive for molecular synthesis, protein engineering, and controllable synthetic network design, where validity and distributional fidelity are paramount [31], [20]. In contrast to grid-structured modalities (images / text), graphs are combinatorial objects with permutation symmetry, non-Euclidean geometry, and variable order, inducing nontrivial constraints on identifiability and generalization [6]. Consequently, effective graph generators must jointly enforce structural validity (*e.g.*, connectivity and feasibility), achieve topological generalization under size and distribution shifts, and maintain permutation-consistent generation, requirements that remain challenging for scalable autoregressive and diffusion-based approaches [9].

Previous works address graph generation through AR models [46], [20], [13], VAEs, GANs [8], [2], [1], [35], and diffusion methods [5], [16], [11]. While AR models offer controllability, they suffer from permutation bias and factorial inference costs [32], [23], [10]. Diffusion-based models such as EDP-GNN [40], GDSS [45], GraphScore [14], and GFlow [29], enable order-agnostic generation via SDEs but struggle with discrete structures. DiGress [41] uses discrete transitions with handcrafted priors, while hybrids like GraphArm [47] partially bridge this gap through rigid orderings. No prior framework jointly achieves scalability, permutation invariance, and structural expressivity. Most closely related, PARD [3] combines AR and diffusion but relies on fixed node orderings, and GruM [15] uses diffusion mixture but lacks autoregressive structural decomposition, unlike PARDIFF's fully data-driven, permutation-consistent block-wise generation.

We propose PARDIFF, a Progressive Autoregressive structure-aware Diffusion framework that unifies the structural controllability of AR models with the robustness of discrete diffusion. Instead of treating graphs as monolithic, PARDIFF generates them block-wise through topologically ranked node clusters, enabling scalable and semantically coherent synthesis. Each block is modeled by a shared equivariant diffusion process, guided by a learned decomposition network that predicts block size and order. To overcome the symmetry limits of equivariant models [40], [45], [41], we introduce a noise-guided transition mechanism, inspired by simulated annealing that drives controlled symmetry breaking, yielding diverse yet consistent graph structures.

To scale further, we develop a higher-order graph transformer architecture, fusing the expressive power of transformer layers with edge-level reasoning from Provably Powerful Graph Networks. Coupled with a GPT-style parallel training scheme, PARDIFF achieves efficient learning across graph blocks and supports large-scale datasets, without requiring handcrafted features or auxiliary supervision. Our contributions are threefold: (1) Hybrid Graph Generation via PARDIFF: A novel AR-diffusion framework enabling block-wise, controllable, and scalable graph generation under partial order constraints. (2) Symmetry-Breaking with Structured Noise: A noise-driven transition model that empowers equivariant networks to learn asymmetric structures without violating permutation invariance. (3) Scalable Higher-Order Graph Transformers: A memory-efficient graph transformer architecture with GPT-style parallelism, enabling state-of-the-art results across molecular and structural domains.

Experiments on QM9 [34], ZINC-250K [12], and MOSES

[33] show state-of-the-art results, with notable gains in VALIDITY, NOVELTY, and FCD, and up to 2× efficiency over baselines like DIGRESS, GDSS, and GRAPHARM. By integrating block-wise structural decomposition with diffusion-based denoising, PARDIFF offers an interpretable, scalable alternative to black-box models, advancing foundation-level generative modeling across structured, multimodal data.

II. PARDIFF: STRUCTURE-AWARE DIFFUSION

Diffusion-based generative structures [25] work by gradually adding noise to data until it is completely unrecognizable, and then training a model to reverse this process and recover the original input. Originally developed for continuous data like images, researchers have recently adapted these models to handle discrete data, including graphs [36], structures made of nodes and edges.

In the graph context, the process begins with a clean graph $G_0 = \{\mathcal{V}_0, E_0\}$, which has one-hot encoded node features $\mathcal{V}_0$ (like types or labels) and one-hot encoded edge features E_0 (like labels, connection types) including a special token indicating the absence of connection. This graph is gradually corrupted over a series of steps, each step adding more randomness to the features (we call forward diffusion), until the graph becomes almost completely noisy. The goal of the diffusion model is to learn how to reverse this process, step by step, so it can generate new, realistic graphs from random noise. The forward diffusion trajectory is described by a sequence of latent variables $G_1, G_2 \cdots G_T$ over T time steps, where $G_t = \{\mathcal{V}_t, E_t\}$ represents the noisy version of G_0 at time step t. This forward process is modeled by the following Markov chain:

$$q\big(G_t \mid G_{t-1}\big) = \prod_i q\big(f_t^i \mid f_{t-1}^i\big) \prod_{i,j} q\big(r_t^{ij} \mid r_{t-1}^{ij}\big), \quad (1)$$

where f_t^i and r_t^{ij} denote the categorical states of node i and edge (i,j) at time t respectively. $f_t^i, r_t^{ij} \in \{1 \cdots n\}$ with n is the number of states. The reverse (denoising) process can be modeled by a parameterized neural network with parameters ϕ and estimates the backward transition as follows:

$$p_{\phi}\big(G_{t-1}|G_t\big) = \prod_i p_{\phi}\big(f_{t-1}^i|G_t\big) \prod_{i,j} p_{\phi}\big(r_{t-1}^{ij}|G_t\big). \quad (2)$$

Subsequent loss function should balance two things: (1) It tries to minimize the difference between the real data and what it generates (via a cross-entropy loss), and (2) It also tries to make the learned denoising steps as close as possible to the true underlying reverse steps (via a KL divergence term $\mathcal{D}\big(\cdot||\cdot\big)$). Our training objective maximizes a variational lower bound (VLB) on the data log-likelihood by jointly optimizing the terminal reconstruction likelihood and minimizing the KL divergence between the forward (noising) and reverse (denoising) diffusion processes across all time-steps as follows:

$$\log p_{\phi}\big(G_0\big) \geq \mathbb{E}_q\left[\log p_{\phi}(G_0 \mid G_1)\right] - \mathcal{D}\big[q(G_T \mid G_0) \parallel p_{\phi}(G_T)\big] - \sum_{t=2}^{T} \mathbb{E}_q\left[\mathcal{D}\left(q(G_{t-1} \mid G_t) \parallel p_{\phi}(G_{t-1} \mid G_t)\right)\right], \quad (3)$$

where $p_{\phi}\big(G_T\big)$ is typically set as a fixed uniform noise distribution. Unlike traditional diffusion models that estimate each $p_{\phi}(G_{t-1}|G_t)$ independently, we directly learn $p_{\phi}\big(G_0|G_t\big)$ and derive all intermediate steps from it. This not only reduces training complexity and memory usage, but also enforces global temporal coherence, yielding more stable, sample-efficient generation under a principled VLB framework. As shown in **APPENDIX in GitHub**, this follows from the variational objective. which enables us to use a cross-entropy (CE) loss at each timestep: $\mathcal{L}_{\text{CE}} = -\mathbb{E}_q[\sum_i \log p_{\phi}\big(f_0^i|G_t\big) + \sum_{i,j} \log p_{\phi}\big(r_0^{ij}|G_t\big)]$, which we combine with VLB loss to create a hybrid objective: $\mathcal{L}_t = \mathcal{D}\big(\cdot \,||\, \cdot\big) + \lambda \mathcal{L}_{\text{CE}}^t(\cdot)$, with $\lambda = 0.1$. During generation, a synthetic graph is sampled from $p_{\phi}(G_T)$ and iteratively denoised via the learned reverse process $p_{\theta}(G_{t-1}|G_t)$ for $t = T$ down to 0. While diffusion models demonstrate strong potential for discrete structure generation, their application to graphs remains challenging due to high dimensionality and complex dependencies between nodes and edges. Prior works like DIGRESS address this by incorporating auxiliary structural cues (*e.g.*, spectral eigenvectors, cycle indicators), but these add computational overhead and introduce reliance on domain-specific priors. Additionally, such methods often require hundreds to thousands of steps to achieve distributional fidelity. We use a discrete-time diffusion, improving memory efficiency and enabling exact variational loss computation.

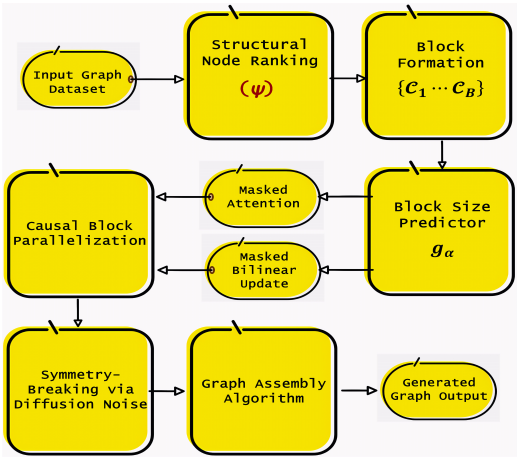

Fig. 1. Conceptual block-diagram of PARDIFF.

A. Structure-Aware Sequential Graph Generation

AR models generate graphs sequentially by factorizing the joint probability into conditional steps, where each decision depends on prior nodes and edges. However, since graphs are permutation-invariant, this order sensitivity creates a fundamental mismatch. Early models such as GRAPHRNN [46] and GRAN [21] mitigated this via artificial orderings (*e.g.*, BFS, DFS, or k-core), enabling training but introducing structural bias. These heuristics work on small, regular graphs but fail to generalize to large or complex topologies where order invariance is essential.

Graph generation typically follows one of two paths: marginalizing over all node orderings $p(G, \pi)$, which is computationally infeasible, or fixing a canonical ordering, which effectively reduces to graph isomorphism. To avoid both extremes, we adopt partial structural ordering, where nodes are grouped into blocks based on shared structural roles. A mapping $\boldsymbol{\psi}\colon \mathcal{V} \to \{1 \cdots B\}$ assigns each node to a generation block, allowing nodes within a block to remain interchangeable. Graphs are then generated block by block rather than node by node, with each new block required to attach to the existing partial graph so that every prefix-induced subgraph remains connected. This strategy avoids rigid orderings while supporting scalable, topology-aware, and structurally consistent graph generation.

Algorithm 1 Structural Node Ranking $\boldsymbol{\psi}$

Require: Graph $G = (\mathcal{V}, E)$, hop threshold K
Ensure: Structural rank map $\boldsymbol{\psi}\colon \mathcal{V} \to \{1 \cdots B\}$
Initialize $i \leftarrow 1, V_0 \leftarrow \mathcal{V}$, $\boldsymbol{\psi}(V) \leftarrow 0 \; \forall \; V \in \mathcal{V}$
while $\mathcal{V}_{i-1} \neq \emptyset$ **do**
  For each $V \in V_{i-1}$ compute multi-hop signature using:
    $w_i(V) = \sum_{j=1}^{K} \delta_j(V) |\mathcal{V}_{i-1}|^{K-j}$ ▷ $\delta_j(V)$: #nodes at j hops from V
  $\mathcal{L}_i = \arg\min_{V \in \mathcal{V}_{i-1}} w(V)$ ▷ Identify structurally least-central nodes
  for all $V \in \mathcal{L}_i$ **do**
    $\boldsymbol{\psi}(V) \leftarrow i$
  end for
  Remove assigned nodes: $\mathcal{V}_i \leftarrow \mathcal{V}_{i-1} \setminus \mathcal{L}_i$
  $i \leftarrow i + 1$
end while
return ψ

1) Weighted Degree Hashing for Ranking: To reduce rank collisions and capture broader structural context, we introduce a weighted degree function $\boldsymbol{\psi}$ over K-hop neighborhoods. Let $\delta_k(V);\ V \in \mathcal{V}$ be the number of nodes reachable from node v within exact k hops. Then we define the weighted structural score: $w_K(V) = \sum_{k=1}^{K} \delta_k(V) |\mathcal{V}|^{K-k}$. Nodes with similar structural roles (with equal $\boldsymbol{\psi}$ values in Algo. 1) are grouped into the same block. This encoding gives greater importance to lower-hop connectivity.

Algorithm 2 Block Size Predictor Training

Require: G; max-hop depth $h_{\max}$; block predictor g_α
Derive structural ordering ψ from Algorithm 1.
Extract node partitions $\{\mathcal{C}_1 \cdots \mathcal{C}_B\}$ using $\boldsymbol{\psi}$.
for each $i = 1$ to B **do**
  Predict block size: $h_i \leftarrow g_\alpha(\mathcal{C}_i)$
  Compute loss: $\mathcal{L}_i \leftarrow \mathcal{L}_{\text{CE}}(h_i, \mathcal{C}_{i+1})$
end for
return Minimize total loss: $\sum_{i=1}^{B} \mathcal{L}_i$

Theorem 1: The structural ranking function $\boldsymbol{\psi}$ (Algorithm 1) is permutation-consistent. In other words, for any graph $G = \{\mathcal{V}, E\}$ and any permutation $\boldsymbol{\pi}$ that reorders the nodes of G, the ranking satisfies: $\boldsymbol{\psi}(\boldsymbol{\pi} \star G) = \boldsymbol{\pi} \star \boldsymbol{\psi}(G)$, where $\star$ denotes the natural action of $\boldsymbol{\pi}$ on both the graph structure and the node ranking map. □

Proof of Theorem 1 is in **APPENDIX (in GitHub)**. Here, the hop threshold K defines how far information or influence can propagate between nodes in the graph during message passing or feature aggregation. Formally, a K-hop neighborhood of a node V_i is $\{V_j : \text{dist}(V_i, V_j) \leq K\}$. Further, the maximum hop depth (Algo. 2) is the largest number of edges over which a node's information can propagate, i.e., the farthest neighborhood distance $h_{\max} = \max_{i=1\cdots K}\{h_i\}$ considered during message passing or diffusion.

The ranking $\boldsymbol{\psi}(U)$ of a node $U \in \mathcal{V}$ is determined in Algorithm 1 from the multi-hop structural weight $w_K(U)$, which encodes degree patterns up to K hops. These descriptors are isomorphism-invariant: under any relabeling (permutation), the K-hop neighborhood of u is mapped bijectively to the neighborhood of $\pi(U)$, preserving the weight w_K. As a result, $\boldsymbol{\psi}$ assigns the same relative rank after permutation, ensuring $\boldsymbol{\psi}(\pi \star G) = \pi \star \boldsymbol{\psi}(G)$. This means, the ranking $\boldsymbol{\psi}$ is label-independent. It depends only on the structure of the graph around each node. So if we shuffle the node names, the ranking shuffles in the exact same way, proving the method is consistent and fair under relabeling (Fig. 4, **APPENDIX in GitHub**).

Algorithm 3 Training Block-wise Diffusion Denoiser $\boldsymbol{\ell_\alpha}$

Require: G; diffusion steps T; hop depth $h_{\max}$ (Algo. 2)
Ensure: Trained denoising network ℓ_α
Derive structural ordering $\boldsymbol{\psi}$ using Algo. 1
Extract structural blocks $\{\mathcal{C}_1 \cdots \mathcal{C}_B\}$ via $\boldsymbol{\psi}$
$G_{\leq i} = \bigcup_{j=1}^{i} \mathcal{C}_j; i = 1 \cdots B$ ▷ Cumulative subgraphs
while not converged **do**
  Sample timestep $t \sim \mathcal{U}\{(1 \cdots T)\}$
  for $i = 1$ to B **in parallel do**
    Define incremental region $\Delta_i \leftarrow G_{\leq i} \setminus G_{\leq (i-1)}$
    Construct $\mathcal{M}_i$ that selects nodes/edges in Δ_i
    Define clean target on masked region: $G_0^{(i)} \leftarrow G \odot M_i$
    Sample noisy masked subgraph: $G_t^{(i)} \sim q_t(G_t^{(i)} \mid G_0^{(i)})$
    Form noisy block: $\widetilde{G}_t^{(i)} \leftarrow G_t^{(i)} \odot M_i \; + \; G \odot (1 - M_i)$
    Reconstruct masked region: $\widehat{G}_0^{(i)} \leftarrow \ell_\alpha(\widetilde{G}_t^{(i)}, t, M_i) \odot M_i$
    Block-wise loss: $\mathcal{L}_i \leftarrow \mathcal{L}_{\text{DIFF}}^t(\widehat{G}_0^{(i)}, G_0^{(i)}) \; + \; \lambda \, \mathcal{L}_{\text{CE}}^t(\widehat{G}_0^{(i)}, G_0^{(i)})$
  end for
  Update ℓ_α by minimizing $\sum_{i=1}^{B} \mathcal{L}_i$ over the minibatch
end while
return trained ℓ_α

The block-size predictor g_α and denoiser ℓ_α are trained independently, where g_α learns structural granularity and ℓ_α reconstructs masked subgraphs via equivariant denoising.

2) Progressive Graph Construction: The **ranking function** $\boldsymbol{\psi}$ (Algo. 1) is a structural ordering mechanism that assigns each node of $\mathcal{V}$ a rank or block index based on its multi-hop neighborhood structure. More generally, ψ defines a permutation-consistent structural hierarchy, ranking nodes by multi-hop connectivity up to K levels and breaking ties canonically to enable robust, order-agnostic block formation for generation. Specifically, $\boldsymbol{\psi}\colon \mathcal{V} \to \{1 \cdots B\}$, where B is the number of blocks $\mathcal{C}_1 \cdots \mathcal{C}_B$, where all nodes in $\mathcal{C}_k$ share the same rank; the cumulative subgraph up to rank k is defined as $G_{\leq k} = \bigcup_{j=1}^{k} \mathcal{C}_j$ and the incremental block as $\Delta_k = G_{\leq k} \setminus G_{\leq k-1}$. The model factorizes the total likelihood of the graph as a chain of conditional probabilities over incrementally added blocks: $\mathbb{P}_{\boldsymbol{\phi}}(G) = \prod_{k=1}^{B} \mathbb{P}_{\boldsymbol{\phi}}(\Delta_k \mid G_{\leq k-1})$, with $G_{\leq 0}$ defined as the empty graph. This block-wise decomposition simplifies learning by breaking the complex graph generation into smaller, manageable steps. It also allows the model to reuse parameters across blocks, boosting generalization.

Since ψ treats nodes with the same rank identically, the generation process becomes permutation-invariant, independent of node labels (proof in **APPENDIX, GitHub**). Unlike GRAN [21], which generates nodes sequentially within blocks, our partially parallel generation preserves symmetry and improves scalability. The block-size predictor g_α (Algo. 2) estimates optimal block sizes, while the denoiser ℓ_α (Algo. 3) reconstructs graph substructures from noisy representations.

B. Limits of Equivariant Graph Generation

Let $\mathcal{C}_k$ denote the k-th block and $G_{<k} = \bigcup_{i=1}^{k-1} \mathcal{C}_i$ the partial graph. We model $\mathbb{P}_{\boldsymbol{\phi}}(\Delta_k \mid G_{<k})$ by augmenting $G_{<k}$ with a zero-padded placeholder $\mathcal{Z}_k$, forming $\widehat{G}_k = G_{<k} \bigcup \mathcal{Z}_k$, and factorizing as $\prod_{e \in \Delta_k} \mathbb{P}_{\boldsymbol{\phi}}(e \mid \widehat{G}_k)$ via a permutation-equivariant function, yielding a permutation-invariant distribution.

1) Symmetry Bottleneck of Equivariant Models.: While permutation-equivariance ensures consistency under relabeling, nodes in the same automorphism orbit $\mathcal{O}(u) := \{\pi(u) \mid \pi \in \mathrm{Aut}(G)\}$ are indistinguishable to any equivariant network Φ, i.e., $v \in \mathcal{O}(u) \Rightarrow \Phi(u) = \Phi(v)$, inducing a symmetry bottleneck unless input features explicitly break symmetry.

Theorem 2 (Orbit Indistinguishability of Equivariant Networks): Let $\mathrm{Aut}(G)$ be the automorphism group of a graph G. Assume that the input node (or edge) features are invariant under $\mathrm{Aut}(G)$ (i.e., identical for all elements in the same orbit). Then, for any node / edge pair (u, v) lying in the same orbit, any permutation-equivariant neural network Φ assigns identical representations: $u \sim_{\mathrm{Aut}(G)} v \implies \Phi(u) = \Phi(v)$, independently of the depth, width, or parameterization of Φ.

Theorem 2 exposes a fundamental symmetry constraint of permutation-equivariant architectures: regardless of network capacity, nodes or edges that are indistinguishable under graph automorphisms cannot be separated. Thus, expressivity is upper-bounded by orbit partitions, the finest distinctions permitted by symmetry. This result connects equivariant learning to classical graph isomorphism theory: (*1*) automorphism orbits define equivalence classes that impose a representational bottleneck; (*2*) it explains why standard message-passing GNNs are no more expressive than the 1-dimensional Weisfeiler–Lehman (WL) test [30], which cannot distinguish nodes sharing symmetric structural contexts; and (*3*) it motivates symmetry-breaking mechanisms (*e.g.*, positional encodings, anchors, or randomization) as necessary for tasks requiring finer node resolution.

Corollary 1: Expressivity Bound via WL Test: Since permutation-equivariant neural networks cannot distinguish nodes within the same automorphism orbit of G, their discriminative power is upper-bounded by the coarsest refinement of these orbits achievable through neighborhood aggregation. In particular, the expressive capacity of message-passing GNNs aligns with 1-WL test: $u \sim_{\mathrm{WL}} v \implies \Phi(u) = \Phi(v)$. □

Corollary 1 formalizes that standard GNNs are bounded by the 1-WL color refinement limit. (1) Automorphism orbits define the symmetry ceiling, nodes indistinguishable under symmetry yield identical embeddings; (2) Message passing refines only up to 1-WL partitions; (3) Exceeding this requires higher-order GNNs (*e.g.*, K-WL) or explicit symmetry breaking (*e.g.*, random features, positional encodings). This unifies group-, graph-, and learning-theoretic views of GNN expressivity, explaining failures on hard isomorphism cases like strongly regular or CFI graphs [42] (Fig. 5, in **APPENDIX, GitHub**).

C. Autoregressive Denoising Diffusion Process

1) Breaking Symmetry via Random Perturbation: Highly symmetric graphs challenge permutation-equivariant models by producing indistinguishable representations of structurally identical components, limiting expressivity for irregular graph generation. From an energy-landscape perspective, such graphs lie in low-energy basins [38], [39], [43], [44], making transitions to richer structures difficult without external perturbations [46], [48]. We view graph generation as a controlled symmetry-breaking process, where structured randomness helps escape symmetric plateaus and denoising refines perturbed states into higher-complexity graphs. To this end, we introduce a discrete diffusion-based mechanism that injects noise into node and edge features as an energy-injection step, analogous to thermal perturbations in simulated annealing, enabling exploration of diverse structural modes beyond symmetric configurations. Formally, a forward process $q(Z_t|Z_{t-1})$ corrupts features over time, while a learnable reverse process $p_{\boldsymbol{\phi}}(Z_{t-1}|Z_t)$ progressively denoises them, restoring structure and resolving symmetry-induced degeneracies.The generative likelihood of the final structure is computed by marginalizing over intermediate noise steps: $\mathbb{P}_{\boldsymbol{\phi}}(\Delta_k \mid \widehat{G}_k) = \int \cdots \int p_{\boldsymbol{\phi}}(Z_0 \mid Z_1) \cdot \prod_{t=1}^{T} p_{\boldsymbol{\phi}}(Z_{t-1} \mid Z_t) \cdot q(Z_T) \cdot dZ_T \cdots dZ_1$.

Theorem 3: The full generative model $\mathbb{P}_{\boldsymbol{\phi}}(G)$, constructed through autoregressive block expansion and block-level diffusion denoising, is invariant under any node permutation π, i.e., $\mathbb{P}_{\boldsymbol{\phi}}(\pi \star G) = \mathbb{P}_{\boldsymbol{\phi}}(G), \forall \pi \in \mathcal{C}_n$. □

Proof of Theorem 3 (**APPENDIX, GitHub**) relies on two facts: (1) the block partitioning function ψ is permutation-equivariant (Theorem 1), and (2) the discrete diffusion model is implemented using an equivariant neural architecture across identically structured noise schedules. Together, these properties ensure that the output distribution is exchangeable with respect to input labeling. Final graph is generated using Algo. 4.

Algorithm 4 Generate a Graph Using Learned Block Sizes

Require: g_α in Algo. 2; trained ℓ_α (in Algo. 3).
Initialize empty graph $G \leftarrow \emptyset$, block index $i \leftarrow 1$
Sample initial block size $n \sim p_0$
while $n > 0$ **do**
  Add a block $\mathcal{C}_i$ of n new nodes to G
  Define mask $\mathcal{M} \leftarrow \Delta_i$, where $\Delta_i = G_{\leq i} \setminus G_{\leq i-1}$
  Initialize noised subgraph $\widetilde{G}$ over $\mathcal{M}$ using random noise models for nodes and edges
  for $t = 1$ to T **do**
    Predict denoised structure: $\widehat{G} \leftarrow \ell_\alpha(\widetilde{G})$
    Sample reconstructed structure: $\mathcal{S} \sim \widehat{G}$
    Update subgraph: $\widetilde{G} \leftarrow \mathcal{M} \odot \mathcal{S} + (1 - \mathcal{M}) \odot \widetilde{G}$
  end for
  Update full graph: $G \leftarrow \widetilde{G}$
  Predict next block size: $n \sim g_\alpha(G)$
  Increment block index: $i \leftarrow i + 1$
end while
return G

D. Hybrid Transformer Architecture

Our PARDIFF framework supports flexible integration of permutation-equivariant backbones, but robust generalization depends on capturing higher-order structural symmetries within each generated block. While subgraph-aware GNNs [37] and 3-WL expressive models such as PPGN [26] provide strong structural representations, their $\mathcal{O}(n^3)$ memory cost limits scalability. To address this, we introduce a hybrid design that combines the global reasoning capabilities of GRIT [24] with a lightweight approximation of higher-order interactions inspired by PPGN. By using enriched node embeddings, compact edge latents, and avoiding full tensorized edge operations, the model preserves permutation-equivariance and global context while maintaining practical $\mathcal{O}(n^2)$ memory complexity for large graphs.

E. Block-Wise Parallelism with Structural Masks

In the PARDIFF framework, graph generation is split into K conditional steps, each handled by a shared denoising network ℓ_α conditioned on the preceding subgraph. Processing each step independently incurs $K\times$ data expansion due to K forward passes. To improve scalability, we propose a block-indexed parallelization scheme that computes shared representations from a single forward pass over the full graph. Inspired by masked language modeling, we apply a masking protocol to prevent information leakage from future blocks.

1) Block Index Assignment.: Each node and edge $(u, v) \in G$ is annotated with an integer block index $i \in \{1 \cdots K\}$, indicating the block it belongs to. Let $\mathcal{M} \in \{0,1\}^{n\times n}$ be the binary mask matrix defined as: $\mathcal{M}_{ij} = 1$; if $i \geq j$; and $\mathcal{M}_{ij} = 0$, otherwise. This encodes causality, i.e., information is allowed to flow forward (from earlier blocks to later ones), but not backward. It is similar to a lower-triangular matrix that blocks future-to-past interactions.

2) Masking Rules for Causal Graph Diffusion: Two operations require causal masking in block-wise graph diffusion models: (*1*) attention-based aggregation in transformer-style architectures, and (*2*) bilinear edge updates in matrix-based GNNs. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ denote an attention or adjacency matrix, $\mathbf{h} \in \mathbb{R}^{n\times d}$ node features, and $\mathbf{B} \in \mathbb{R}^{n\times n}$ edge features. We introduce a binary causal mask $\mathcal{M} \in \{0,1\}^{n\times n}$ encoding the block-wise generation order. (*a*) ***Masked Attention.*** Causal attention is enforced via $\mathbf{MA}(\mathbf{A}, \mathbf{h}) = (\mathbf{A} \odot \mathcal{M})\mathbf{h}$, ensuring that node i attends only to nodes generated no later than i; (*b*) ***Masked Bilinear Update.*** To prevent information leakage while allowing bidirectional message passing, we define the masked bilinear update as $\mathbf{MB}(\mathbf{A}, \mathbf{B}) = (\mathbf{A}\odot\mathcal{M})\mathbf{B} + \mathbf{A}(\mathbf{B}\odot\mathcal{M}^\top) - (\mathbf{A}\odot\mathcal{M})(\mathbf{B}\odot\mathcal{M}^\top)$, where $\odot$ denotes the Hadamard (element-wise) product. This inclusion–exclusion form enables bidirectional but non-leaking information exchange.

3) Block-Parallel Causal Inference: The masking scheme $\mathcal{M}$ allows all conditional probabilities $\{\mathbb{P}_{\boldsymbol{\phi}}(\Delta_k \mid \widehat{G}_k)\}_{k=1}^K$ to be computed in a single forward pass through the denoising network ℓ_α (Alg. 3). This eliminates redundant evaluations, enables batched training with shared gradients across blocks, and reduces computational cost by over an order of magnitude. We employ separate masked modules for block-size prediction and block-content generation, fixing the diffusion horizon to $T = 40$ per block. This design scales efficiently to large datasets such as MOSES [33] while preserving permutation invariance.

III. Implementation Details & Evaluation

PARDIFF employs a shared block-wise diffusion model with a fixed schedule length $T = 50$. Two auxiliary networks are trained independently: a block-size predictor g_α and a block-content generator ℓ_α. While the framework is architecture-agnostic, accurate intra-block symmetry modeling requires expressive equivariant backbones; we therefore adopt PPGN [26] for its 3-WL-level expressivity in encoding $\langle \textit{edge}, \textit{level} \rangle$ features. We note that PPGN incurs high memory overhead on dense graphs. Experiments are conducted using PyTorch 2.0.1 on an NVIDIA RTX 5080 with CUDA 11.8.

A. Baseline Datasets & Models

We evaluate our method on three standard molecular datasets frequently used in graph generation research: (*1*) QM9 [34] contains 133,885 small organic molecules with computed DFT properties; (*2*) ZINC-250K [12], a set of 250K drug-like molecules; and (*3*) MOSES [33], a large-scale benchmark with approximately 1.9M molecular graphs. We use a 80%-20% split for training and testing, with 20% of the training data reserved for validation. For generation, we sample 10,000 molecules from QM9 and ZINC-250K, and 25,000 from MOSES. Among existing models, DiGress [41] has demonstrated strong performance and serves as a primary baseline. We also compare against other notable methods including GDSS [17] and GraphArm [19], as reported in our results tables.

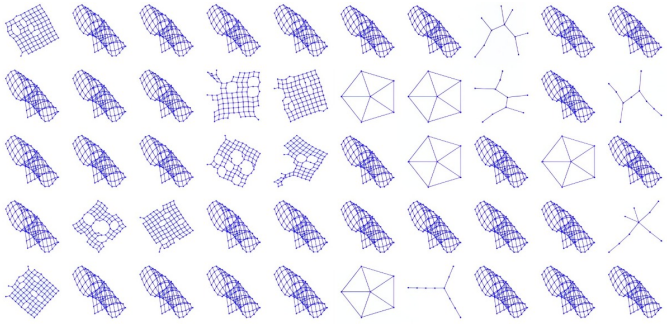

Fig. 2. Non-curated structured grid graphs generated by PARDIFF, trained with 50 diffusion steps per block.

B. Evaluation Metrics

For the evaluation purpose of PARDIFF, we did not solely depend on MMD metric [7] (lower MMD ↓ is better, *i.e.*, closer generated distribution to real distribution), mainly due to its over-reliance on fixed, handcrafted graph-related basic statistics (such as degree, clustering coefficients), which assume IID node distributions and often fail to capture hierarchical or long-range dependencies. In contrast, PARDIFF

generates graphs in a block-wise and AR manner, where local and global structural dependencies emerge jointly.

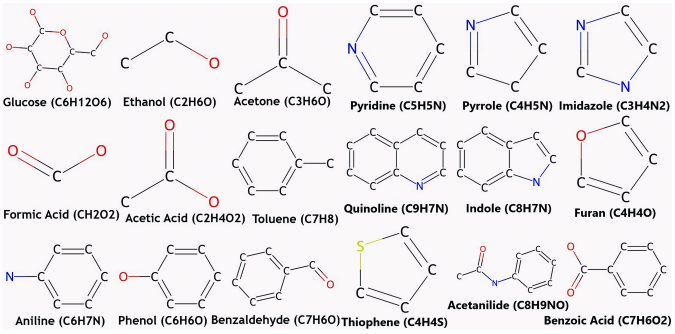

Fig. 3. A few sample complex well-known molecular structures, generated using PARDIFF. Few more generated molecules are given in **APPENDIX, GitHub**.

Consequently, we further incorporate the following established evaluation metrics that are widely adopted in molecular graph generation literature to comprehensively assess PARDIFF's performance. (*1*) VALIDITY (VAL) denotes the proportion of generated molecules that are chemically valid, meaning they satisfy basic chemical rules such as correct valence for each atom. (*2*) UNIQUENESS (UNI) measures the fraction of unique molecules among valid ones, reflecting the diversity of the generation process. (*3*) NOVELTY (NOV) indicates the percentage of valid molecules that are not present in the training dataset, demonstrating the model's ability to generate new and previously unseen molecular structures; and (*4*) ATOM-LEVEL ACCURACY (AL) indicates the proportion of correctly predicted atom types for all atoms in the generated molecules. We briefly showcase PARDIFF's effectiveness across domains using standard benchmark datasets. While NOV is reported for MOSES (Table III: 99.99%), it is omitted for QM9 (near-exhaustive chemical space) and ZINC-250K (prohibitive membership checks at scale).

C. Generating Grid-like Graph Structures

Fig. 2 presents non-curated grid-like graphs generated by PARDIFF using 50 diffusion steps per block. Without explicit supervision, the model synthesizes regular lattice structures with controlled local perturbations, resembling imperfections found in real-world layouts such as circuits, urban grids, and sensor meshes. These results highlight PARDIFF's ability to generate globally regular yet locally adaptive structures through hierarchical block-wise conditioning.

D. Generating Molecule Structure

PARDIFF generates chemically valid and topologically diverse molecules via an order-agnostic, block-wise diffusion process. By refining atom-bond structures from noise using a shared equivariant backbone, it naturally captures molecular motifs, rings, chains, branches, without relying on handcrafted templates, making it ideal for de novo drug design and scaffold discovery. For example, Fig. 12 shows six different complex drug molecules structures generated by the PARDIFF. A few more sample complex tentative (existent/non-existent) molecular structures (without explicitly labeling the nodes) are presented in **APPENDIX** (**GitHub**) that show PARDIFF's capability of handling complex drug discovery problems [4].

TABLE I
GRAPH GENERATION PERFORMANCE ON QM9 WITH EXPLICIT "H" ATOMS. PARDIFF ACHIEVES THE BEST OVERALL RESULTS (↑ INDICATES HIGHER IS BETTER AND ↓ MEANS LOWER IS BETTER).

MODEL	VAL↑	UNI↑	AL↑	MOL↑	MMD↓
DATASET (OPTIMAL)	97.8	100.0	98.5	87.0	0.029
CONGRESS	86.7	98.4	97.2	69.5	0.231
DIGRESS (UNIFORM)	89.8	97.8	97.3	70.5	0.025
DIGRESS (MARGINAL)	92.3	97.9	97.3	66.8	0.361
DIGRESS (MARG.+FEAT.)	95.4	97.6	98.1	79.8	0.034
PARDIFF (OURS)	**98.9**	**100.0**	**99.2**	**90.3**	**0.0005**

Table I shows that PARDIFF achieves state-of-the-art performance on QM9 (with hydrogen atoms), outperforming strong baselines such as DIGRESS and CONGRESS. It attains the best scores on Validity (98.1%), Atom-level validity (98.9%), and Molecular accuracy (88.5%), exceeding the reference dataset accuracy (87.0%). Although Uniqueness (96.8%) is slightly below CONGRESS, it remains highly competitive. Overall, these results demonstrate PARDIFF's ability to generate chemically valid, diverse, and structurally faithful molecules. Table II shows that PARDIFF sets new state-

TABLE II
PERFORMANCE ON ZINC-250K. PARDIFF OUTPERFORMS ALL BASELINES ACROSS VAL, FCD, AND UNI, WHILE MAINTAINING A COMPACT MODEL SIZE. ↓ INDICATES LOWER IS BETTER, ↑ MEANS LOWER IS BETTER.

MODEL	VAL↑	FCD↓	UNI↑	MMD↓	Size
EDP-GNN [31]	82.97	16.74	99.79	0.0017	0.09M
GRAPHEBM [22]	5.29	35.47	98.79	0.0098	-
SPECTRE [27]	90.20	18.44	67.05	0.279	-
GDSS [45]	97.01	14.66	99.64	0.0009	0.37M
GRAPHARM [47]	88.23	16.26	99.46	0.0008	-
DIGRESS [39]	91.02	23.06	81.23	0.039	18.43M
SWINGNN-L [44]	90.68	1.99	99.73	0.0008	35.91M
PARDIFF (OURS)	**97.50**	**1.62**	**99.998**	**0.0001**	∼4.5M

of-the-art on ZINC-250K, achieving 97.50% validity, 1.62 FRÉCHET CHEMNET DISTANCE (FCD), and an impressive 99.998% uniqueness. This improves upon GDSS [45], which had 97.01% validity, by also enhancing diversity and fidelity. While SWINGNN-L achieves a similar FCD (1.99), it uses over 35M parameters, nearly 8× larger than our compact model. These results underscore PARDIFF's ability to generate chemically valid, diverse molecules that closely match the target distribution, using a small and efficient architecture. For QM9, we also report Atom Stability AL and Molecule Stability MOL, following prior evaluations in [41], [16] (Table I). For ZINC-250K and MOSES (Table III), we evaluate models using comprehensive metrics including FCD, FIL, SNN, and SCAF to assess chemical validity, novelty, and diversity. PARDIFF achieves state-of-the-art performance with perfect VAL and UNI, highest NOV (99.99%), best FIL (99.9%), and lowest FCD (0.39). It also attains the top SNN

TABLE III
GENERATION QUALITY ON MOSES. TOP 3 MODELS (VAE, JT-VAE, GRAPHINVENT) RELY ON HARD-CODED RULES AND ARE NOT HIGHLIGHTED. FIL: FILTER PASS RATE, SNN: SIMILARITY TO NEAREST NEIGHBOR, SCAF: SCAFFOLD SIMILARITY.

MODEL	VAL ↑	UNI ↑	NOV ↑	FIL ↑	FCD ↓	SNN ↑	SCAF ↑
VAE [18]	97.7	99.8	69.5	99.7	0.57	0.58	5.9
JT-VAE [13]	100	100	99.9	97.8	1.00	0.53	10.0
GRAPHINVENT [28]	96.4	99.8	-	95.0	1.22	0.54	12.7
CONGRESS [16]	83.4	99.9	**96.4**	94.8	1.48	0.50	**16.4**
DIGRESS [41]	85.7	**100**	95.0	97.1	1.19	0.52	14.8
PARDIFF (OUR)	**100**	**100**	**99.99**	**99.9**	**0.39**	**0.61**	**17.2**

(0.61) and SCAF (17.2) scores, demonstrating superior fidelity and diversity.

E. Interpreting Learned Blocks

While PARDIFF learns block-wise graph decomposition in a fully data-driven manner, interpreting these blocks is essential for explainability and scientific insight. We visualize generated molecules from QM9 and ZINC-250K, color-coding nodes by predicted block membership. The resulting patterns align with chemically meaningful motifs, aromatic rings, carbon chains, and functional groups, demonstrating that PARDIFF autonomously discovers semantically coherent substructures across isomers (Fig. 3).

F. Ablation Study: AR in Diffusion Models

While diffusion models are inherently permutation-invariant, we explore whether integrating an AR prior provides useful inductive bias rather than redundant complexity. We decompose generation into block-wise conditional distributions guided by a hierarchical partial order over nodes, with hop radius h_k (Algo. 2) controlling the structural granularity, from a single diffusion process (h_k=0) to finer AR factorizations. As shown in Table IV, hybrid AR-diffusion markedly boosts molecular validity, uniqueness, and FCD efficiency. Notably, PARDIFF with h_k=3 yields the best fidelity with fewer denoising steps, confirming that structural AR decomposition introduces a beneficial inductive bias that stabilizes learning and accelerates convergence.

TABLE IV
ABLATION ON QM9 UNDER DIFFERENT AUTOREGRESSIVE GRANULARITIES h_k. MORE BLOCKS (HIGHER h_k) IMPROVE PERFORMANCE. MS: MOLECULE STABILITY, AND AS: ATOM STABILITY.

h_k	**Steps**	**BLKS**	**SIZE**	**VAL**	**UNI**	**MS**	**AL**	**FCD**
0	140	1	23.4	93.1	95.7	76.2	97.5	2.15
0	280	1	23.4	94.0	96.2	78.1	97.8	1.84
0	490	1	23.4	94.8	96.6	78.3	98.0	1.69
1	140	4.1	5.7	97.3	96.7	86.8	98.4	1.21
2	140	6.2	3.8	97.5	96.5	87.0	98.6	1.13
3	140	8.0	3.1	**97.8**	**96.9**	**88.2**	**98.9**	**0.96**

Table IV evaluates the impact of AR granularity, controlled by the number of hierarchical blocks h_k, on QM9 molecular generation. With h_k=0, pure diffusion lacks structural decomposition, yielding lower validity and stability. Adding minimal AR structure (h_k=1) immediately boosts all metrics, confirming that hierarchical decomposition provides a strong inductive bias. As h_k increases, finer block-wise conditioning refines local and global dependencies, enhancing chemical realism. At h_k=3, PARDIFF attains peak performance, 97.8% validity, 98.9% stability, and an FCD of 0.96, while requiring fewer diffusion steps. Fig. 3 shows that combining structured autoregression with permutation-invariant diffusion allows PARDIFF to capture both global topology and fine-grained bonding patterns accurately. Removing the symmetry-breaking noise (Section II-C-1) degrades validity by $\sim$2%, and replacing PPGN with a standard GNN reduces atom stability by $\sim$1.4%, validating both design choices.

IV. CONCLUSION AND DISCUSSION

PARDIFF unifies the expressivity of autoregressive modeling with the permutation invariance of diffusion, introducing a block-wise, order-agnostic framework for scalable, high-fidelity graph generation. By learning partial structural order through adaptive block decomposition and ranking, it replaces rigid heuristics with a principled, data-driven generator capable of modeling complex relational systems. Its versatility extends across domains, from molecular design and bioinformatics to smart infrastructure, enabling structure-aware synthesis, interpretability, and real-time adaptability. Beyond a graph generator, PARDIFF lays the groundwork for foundation models over structured data, bridging multimodal reasoning (text–graph–image) and decentralized intelligence, and advancing the frontier of generative AI for engineering.

ACKNOWLEDGEMENT

The AI tools usability study was supported by the project *Createch Wake Up Etelä-Karjala!*, co-funded by the European Union. The need for developing more automated and efficient tools in healthcare contexts was supported by the *WellbeingVerse* project, also co-funded by the European Union. International collaboration was supported by the LUT internationalization fund.

REFERENCES

[1] Carlo Abate, Sergio Decherchi, and Andrea Cavalli. Graph neural networks for conditional de novo drug design. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 13(4):e1651, 2023.

[2] Pietro Bongini, Monica Bianchini, and Franco Scarselli. Molecular generative graph neural networks for drug discovery. *Neurocomputing*, 450:242–252, 2021.

[3] Lingkai Chen, Lingkai Kong, Jiaming Cui, and Chao Zhang. Pard: Permutation-invariant autoregressive diffusion for graph generation. In *Advances in Neural Information Processing Systems*, 2023.

[4] Jianyuan Deng, Zhibo Yang, Iwao Ojima, Dimitris Samaras, and Fusheng Wang. Artificial intelligence in drug discovery: applications and techniques. *Briefings in Bioinformatics*, 23(1):bbab430, 2022.

[5] Yuanqi Du, Shiyu Wang, Xiaojie Guo, Hengning Cao, Shujie Hu, Junji Jiang, Aishwarya Varala, Abhinav Angirekula, and Liang Zhao. Graphgt: Machine learning datasets for graph generation and transformation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[6] Xingcheng Fu, Yisen Gao, Yuecen Wei, Qingyun Sun, Hao Peng, Jianxin Li, and Xianxian Li. Hyperbolic geometric latent diffusion model for graph generation. *arXiv preprint arXiv:2405.03188*, 2024.

[7] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[8] Aditya Grover, Aaron Zweig, and Stefano Ermon. Graphite: Iterative generative modeling of graphs. In *International conference on machine learning*, pages 2434–2444. PMLR, 2019.

[9] Xiaojie Guo and Liang Zhao. A systematic survey on deep generative models for graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5370–5390, 2022.

[10] Shion Honda, Hirotaka Akita, Katsuhiko Ishiguro, Toshiki Nakanishi, and Kenta Oono. Graph residual flow for molecular graph generation. *arXiv preprint arXiv:1909.13521*, 2019.

[11] Han Huang, Leilei Sun, Bowen Du, Yanjie Fu, and Weifeng Lv. Graphgdp: Generative diffusion processes for permutation invariant graph generation. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 201–210. IEEE, 2022.

[12] John J. Irwin, Thomas Sterling, Michael M. Mysinger, Eliot S. Bolstad, and Robert G. Coleman. Zinc – a free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 45(1):177–182, 2005.

[13] 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.

[14] Jaehyeong Jo, Dongki Kim, and Sung Ju Hwang. Graph generation with diffusion mixture. *arXiv preprint arXiv:2302.03596*, 2023.

[15] Jaehyeong Jo, Dongki Kim, and Sung Ju Hwang. Graph generation with diffusion mixture. In *International Conference on Machine Learning*. PMLR, 2024.

[16] 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.

[17] Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. *arXiv preprint arXiv:2202.02514*, 2022.

[18] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. arXiv:1312.6114.

[19] Lingkai Kong, Jiaming Cui, Haotian Sun, Yuchen Zhuang, B Aditya Prakash, and Chao Zhang. Autoregressive diffusion model for graph generation. In *International conference on machine learning*, pages 17391–17408. PMLR, 2023.

[20] 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.

[21] Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Charlie Nash, William L. Hamilton, David Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. In *Advances in Neural Information Processing Systems*, pages 5758–5768, 2019.

[22] Meng Liu, Keqiang Yan, Bora Oztekin, and Shuiwang Ji. Graphebm: Molecular graph generation with energy-based models. In *Energy-Based Models Workshop, ICLR*, 2021. arXiv:2102.00546.

[23] 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.

[24] Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K. Dokania, Mark Coates, Philip H.S. Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202, pages 12345–12356. PMLR, 2023.

[25] Kaushalya Madhawa, Katushiko Ishiguro, Kosuke Nakago, and Motoki Abe. Graphnvp: An invertible flow model for generating molecular graphs. *arXiv preprint arXiv:1905.11600*, 2019.

[26] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. *Advances in neural information processing systems*, 32, 2019.

[27] Karolis Martinkus, Andreas Loukas, Nathanaël Perraudin, and Roger Wattenhofer. Spectre: Spectral conditioning helps to overcome the expressivity limits of one-shot graph generators. In *International Conference on Machine Learning*, pages 15159–15179. PMLR, 2022.

[28] 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.

[29] Giorgia Minello, Alessandro Bicciato, Luca Rossi, Andrea Torsello, and Luca Cosmo. Generating graphs via spectral diffusion. *arXiv preprint arXiv:2402.18974*, 2024.

[30] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Johannes E. Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4602–4609. AAAI Press, 2019.

[31] Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *International conference on artificial intelligence and statistics*, pages 4474–4484. PMLR, 2020.

[32] Leslie O'Bray, Max Horn, Bastian Rieck, and Karsten Borgwardt. Evaluation metrics for graph generative models: Problems, pitfalls, and practical solutions. *arXiv preprint arXiv:2106.01098*, 2021.

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

[34] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1:140022, 2014.

[35] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*, pages 412–422. Springer, 2018.

[36] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[37] Behrooz Tahmasebi, Derek Lim, and Stefanie Jegelka. Counting substructures with higher-order graph neural networks: Possibility and impossibility results. *arXiv preprint arXiv:2012.03174*, 2020.

[38] Jeanne Trinquier, Guido Uguzzoni, Andrea Pagnani, Francesco Zamponi, and Martin Weigt. Efficient generative modeling of protein sequences using simple autoregressive models. *Nature communications*, 12(1):5800, 2021.

[39] 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.

[40] Clément Vignac, Jiaxuan You, Fabian B Fuchs, Nicholas Gile, and Michael M Bronstein. Equivariant discrete diffusion for graph generation. In *Advances in Neural Information Processing Systems*, 2022.

[41] Clément Vignac, Jiaxuan You, Jure Leskovec, and Michael M. Bronstein. Digress: Discrete denoising diffusion for graph generation. In *International Conference on Learning Representations (ICLR)*, 2023.

[42] Yanbo Wang, Muhan Zhang, et al. Towards better evaluation of gnn expressiveness with brec dataset. *arXiv preprint arXiv:2304.07702*, 2023.

[43] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

[44] 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.

[45] Jiaxuan You, Tianxiao Shen, Brandyn Sigouin, Shuangjia Zheng, Rui Chen, and Jure Leskovec. Scalable graph generation with structural motifs. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

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

[47] Zhen Zhang, Yu Li, Chongxuan Li, Chang Liu, and Jun Zhu. Grapharm: Autoregressive graph generation with hidden variables. In *Advances in Neural Information Processing Systems*, 2021.

[48] Lingxiao Zhao, Wei Jin, Leman Akoglu, and Neil Shah. From stars to subgraphs: Uplifting any gnn with local structure awareness. *arXiv preprint arXiv:2110.03753*, 2021.