G2SAE: SPECTRALLY SUPERVISED GRAPH-LEVEL AUTOENCODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph-level autoencoding aims to compress graphs into graph-level latent representations from which their pairwise connectivity and node-level attributes can be recovered. Learning such representations requires preserving structural information while accounting for arbitrary node orderings during reconstruction supervision. We introduce G2SAE, a graph-level autoencoder that uses an explicitly supervised Laplacian spectral representation to guide graph reconstruction and node alignment. The encoder derives a spectral latent from graph topology and uses topological features to condition separate node and edge semantic latents. The decoder predicts node-level spectral coordinates from the spectral latent, then combines these coordinates and their pairwise relations with the semantic latents to reconstruct connectivity and attributes. The spectral representation also provides structural cues to align predicted and input nodes, enabling spectral and graph reconstruction objectives to supervise both the intermediate representation and the recovered graph. Experiments on molecular graph benchmarks demonstrate improved reconstruction fidelity, highlighting the effectiveness of G2SAE in learning graph-level representations that retain structural and semantic information.

1 INTRODUCTION

Graph representation learning provides a common framework for modeling relational data in domains such as social networks, chemistry, and biology (Zhang et al., 2022; Ju et al., 2024). For graph-level tasks such as classification and regression, representations are expected to keep useful information for prediction, but do not need to support reconstruction of the original graph. For example, InfoGraph (Sun et al., 2020) and GraphMAE (Hou et al., 2022) learn graph representations for downstream tasks without requiring complete graph reconstruction from the final embedding. However, Graph-level autoencoding proposes a different requirement: a compact, fixed-dimensional latent representation must support recovery of graph size, connectivity, and node and edge attributes. Therefore, reconstruction provides a direct way to test what information can still be recovered after compression. This encoder-decoder mapping also provides a basis for graph generation and latent space optimization, allowing continuous representations to be decoded into discrete graph structures (Jin et al., 2018; Zhang et al., 2019).

This goal differs from node-level autoencoding. In the variational graph autoencoder (VGAE) (Kipf & Welling, 2016), each node retains its own latent variable, and the connections between nodes are reconstructed from pairs of these latent variables. Simply pooling these node representations does not train a decoder to recover the complete graph from the pooled vector. With a graph-level bottleneck, the decoder must recreate node-wise representations and their relations without relying on the original node indices. Because a graph can have multiple equivalent node orderings, reconstruction must be evaluated under node permutations rather than by comparing fixed matrix positions (Yang et al., 2019). The objective is to recover an isomorphic graph, not the original node ordering.

Existing graph-level autoencoders address this correspondence problem in different ways. GraphVAE (Simonovsky & Komodakis, 2018) uses approximate graph matching to align predictions and targets. PIGVAE (Winter et al., 2021) learns a permutation. GRALE (Krzakala et al., 2025) incorporates higher-order structural information, including shortest paths, into its graph features and recon-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

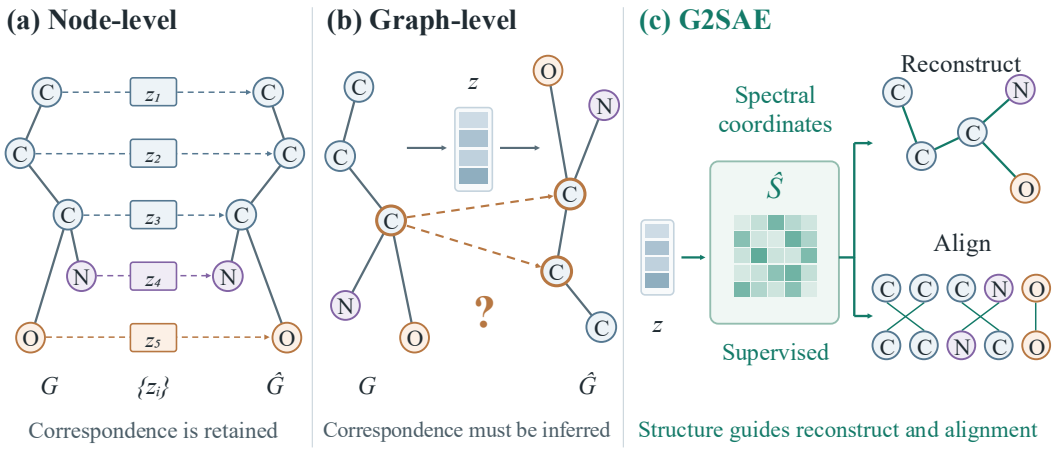


Figure 1: Comparison of node-level and graph-level autoencoding paradigms. (a) Node-level latents retain correspondence with the original nodes. (b) A graph-level bottleneck does not preserve direct node correspondence, so alignment must be inferred. (c) G2SAE recovers supervised spectral coordinates to guide both graph reconstruction and node alignment.

struction targets. Its matcher reuses encoder and decoder states to learn node affinities. However, the hidden states used for matching are not directly supervised to encode fine-grained node-level descriptors with explicit structural meaning. So their structural interpretation remains implicit.

This distinction is important because graph reconstruction and node alignment make related requirements on the decoded representation. Graph reconstruction requires recovering connectivity as well as node and edge attributes, whereas graph matching can benefit from node representations that capture structural context (Wang et al., 2019). For example, carbon atoms can share the same category while occupying different structural positions within a molecule (Dwivedi et al., 2022). Therefore, a category consistent but structurally inconsistent correspondence can misalign edge-level reconstruction targets. A structural target for the decoded node representations could provide a common basis for both operations, supplementing supervision of the final graph. This leads to a natural question: *Can a graph-level autoencoder recover a node representation with an explicit structural definition that directly supports both graph decoding and node correspondence?* As shown in Fig. 1, our approach makes this representation an explicit reconstruction target and uses it to guide the recovery of connectivity and attributes.

Laplacian spectral coordinates provide such a structural representation. Derived from graph topology, they assign a structural coordinate vector to each node. SPECTRE (Martinkus et al., 2022) demonstrates the value of conditioning graph generation on Laplacian eigenvalues and eigenvectors. Our goal is to recover structural spectral coordinates, constructed from Laplacian eigenvalues and eigenvectors, from the latent representation of a specific input graph. These coordinates can be used to construct the graph Laplacian, linking node-level structural information to pairwise connectivity. The supervision targets are derived from the input topology without external annotations. These coordinates provide structural cues for both reconstruction and matching, rather than a unique canonical node ordering. Node and edge semantics complement this structural representation when reconstructing attributed graphs.

We introduce G2SAE, a graph-level autoencoder built around supervised spectral representation. The encoder extracts topological features and uses them to guide semantic encoding. Separate pooling operations then produce graph-level latent representations for structure, node semantics, and edge semantics. The decoder receives only these graph-level latent representations, rather than the node representations directly from the encoder. The structural latent representation is decoded into spectral coordinates, whose pairwise relations, together with the semantic latent representations, guide the joint reconstruction of connectivity and attributes. Consequently, spectral prediction is an integral part of the decoding path rather than merely an auxiliary output. During training, a matcher module combines spectral coordinate and node category costs, then refines them using adjacency

responses under spectral probes. This process introduces neighborhood consistency without explicitly enumerating dense edge-to-edge correspondence costs. Spectral and graph reconstruction are supervised based on the derived node correspondence. Padding nodes are treated as an additional node category and fully participate in the whole pipeline, allowing graph size to be determined from the predicted non-padding nodes. Experiments on molecular graph benchmarks demonstrate improvements in graph edit distance and exact reconstruction accuracy. These results support the use of supervised spectral representations for faithful graph-level reconstruction.

In summary, our main contributions are as follows:

- Develop a graph-level autoencoder that recovers explicitly supervised spectral coordinates from a structural latent, providing a shared structural representation for graph reconstruction and node alignment.
- Design topology-conditioned encoding and decoding with structural, node-semantic, and edge-semantic graph-level latents, using predicted spectral coordinates and pairwise relations to jointly recover connectivity and attributes.
- Propose a parameter-free node matcher combining spectral coordinates, node categories, and neighborhood probes to establish correspondence without explicitly constructing dense edge-to-edge matching costs.

2 Related Work

2.1 Graph Latent Space Learning

Graph latent representation learning includes embeddings of individual nodes and global representations of entire graphs. At the node level, Kipf & Welling (2016) propose VGAE, which uses a GCN encoder and an inner product decoder to reconstruct adjacency from node latent variables. Grover et al. (2019) propose Graphite, which extends this setting with iterative decoding and refines graph estimates through repeated message passing. At the graph level, Sun et al. (2020) introduce InfoGraph, which learns global representations by maximizing mutual information between graph and substructure embeddings at different scales. Hou et al. (2022) reconstruct masked node attributes in GraphMAE and pool the learned node embeddings for graph-level tasks. Therefore, InfoGraph and GraphMAE can learn graph-level representations for downstream tasks, but they do not require the intermediate graph embeddings to reconstruct the complete input graph. For molecular generation, Jin et al. (2018) propose Junction Tree VAE, which first decodes a tree of valid substructures and then assembles the full molecular graph.

G2SAE learns separate graph-level latent vectors for structure and semantic information, and trains them through reconstruction of graph size, connectivity, and attributes.

2.2 Graph-Level Autoencoding and Graph Matching

Graph-level autoencoding must reconstruct a complete graph from a compact, permutation-invariant latent representation, while reconstruction requires a permutation-consistent correspondence between predicted and target nodes. Simonovsky & Komodakis (2018) propose GraphVAE, which decodes a probabilistic graph under a maximum size constraint and applies approximate graph matching before computing the reconstruction loss. Winter et al. (2021) learn a permutation between input and decoded node orderings through a continuous relaxation of sorting, replacing explicit matching optimization for each sample. They also predict graph size from the latent representation during generation. Krzakala et al. (2025) jointly train an Evoformer-based autoencoder and a differentiable matcher in GRALE. The matcher converts learned node affinities into soft correspondences with Sinkhorn normalization, and the reconstruction loss compares node existence, attributes, and connectivity under this correspondence. In BrepVGAE, Guo et al. (2026) represent B-rep faces and edges as graph nodes and compress their geometry and topology into one global latent vector. Its set decoder uses learnable queries and cross-attention to recover node features in parallel, while bilinear layers reconstruct graph connectivity.

G2SAE instead uses supervised spectral coordinates as structural signals for node matching, combine with predicted node types to establish node correspondences.

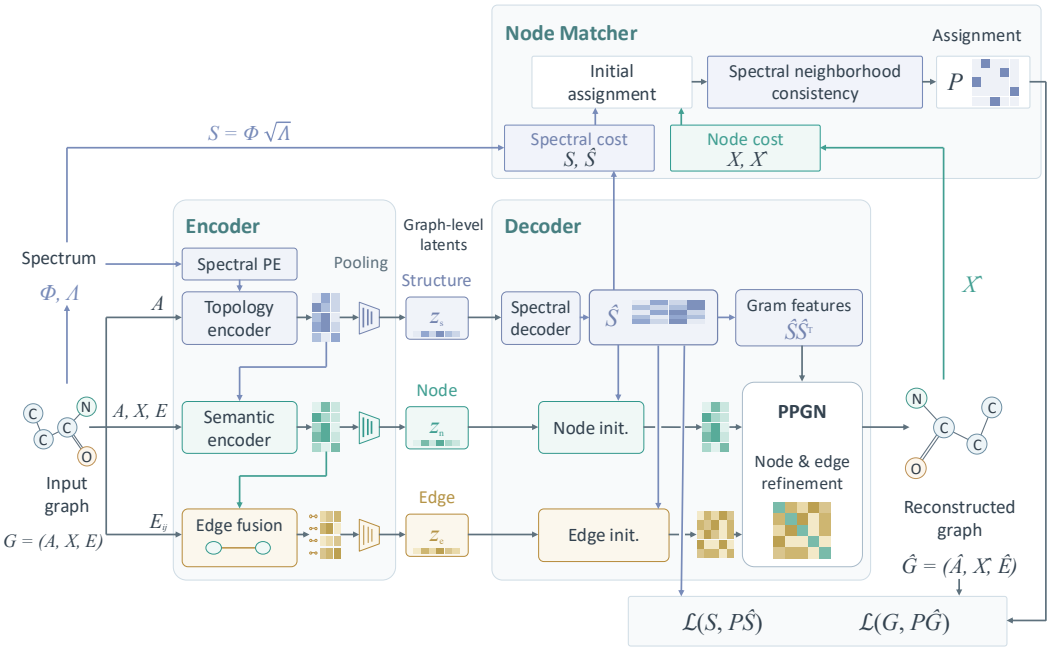


Figure 2: G2SAE architecture. The encoder produces structural and semantic graph-level representations. Predicted spectral coordinates guide graph reconstruction and, together with node type information, support node matching. The resulting correspondences are used to compute spectral and graph reconstruction losses against their respective targets.

2.3 Graph Spectral Representations

Graph spectra provide both global and local structural information. Laplacian eigenvalues encode global graph properties (Shuman et al., 2013), while eigenvectors can be used to construct node coordinates (Belkin & Niyogi, 2003). Dwivedi & Bresson (2021) use Laplacian eigenvectors as positional encodings in their Graph Transformer, while Rampášek et al. (2022) include spectral encodings within a broader set of positional and structural features in GraphGPS. Lim et al. (2023) introduce SignNet and BasisNet to handle sign ambiguity and basis ambiguity in Laplacian eigenvectors. Martinkus et al. (2022) generate Laplacian eigenvalues and eigenvectors in SPECTRE before using them to condition graph generation. Minello et al. (2025) model the joint distribution of eigenvalues and eigenvectors with diffusion in GGSD, then use a PPGN predictor to recover adjacency from the generated spectral representation. Hermanns et al. (2021) use Laplacian spectral signatures in GRASP to establish correspondence between observed graphs.

G2SAE reconstructs supervised spectral coordinates from its structural graph latent and uses them as an intermediate representation for attributed graph reconstruction.

3 G2SAE Architecture

G2SAE is a graph-level autoencoder built around an explicitly supervised spectral representation that supports both graph reconstruction and node alignment. As illustrated in Fig. 2, the encoder first extracts topological features and uses them to condition semantic encoding, generating graph-level latent representations for structure, node semantics, and edge semantics, respectively. The decoder recovers spectral coordinates from the structural latent representation and combines these coordinates and their pairwise relations with the semantic latent representations to jointly reconstruct connectivity and attributes. These coordinates also guide node matching, enabling the spectral representation and the reconstructed graph to be supervised under a unified node correspondence.

3.1 Topological Conditioned Graph Encoder

We represent a graph as $G = (A, X, E)$, with adjacency A and node and edge attributes X and E. Graphs are padded to $N = N_{\max}$ nodes, with padding nodes treated as an additional node category. Laplacian eigenvectors provide structural coordinates for nodes (Belkin & Niyogi, 2003; Martinkus et al., 2022). We retain the first K eigenvectors Φ_K and corresponding eigenvalues Λ_K of the graph Laplacian $L = D - A$ in increasing eigenvalue order, where D is the degree matrix. Their scaled product $S = \Phi_K\sqrt{\Lambda_K}$ defines our spectral reconstruction target.

Inspired by spectral attention (Kreuzer et al., 2021), we use a hybrid spectral positional encoding that combines eigenvalue attention with MLP gating to modulate the corresponding eigenvectors. The encoding is fed into a GIN topology encoder (Xu et al., 2019), whose output provides structural context for a GINE semantic encoder (Hu et al., 2020):

$$\begin{aligned} H_{\text{topo}} &= \text{GIN}(A, \text{PE}(\Phi_K, \Lambda_K)), \\ H_{\text{node}} &= \text{GINE}(A, E, [X \| H_{\text{topo}}]). \end{aligned} \tag{1}$$

Here $\|$ denotes concatenation. This hierarchy encodes attributes in their structural context before graph-level compression.

We pool the topological and node-semantic representations separately. For edge semantics, gated aggregation combines edge types with their semantic endpoint representations into node-wise edge contexts H_{edge}. The resulting graph-level latent representations are

$$z_s = \text{AttnPool}_s(H_{\text{topo}}), \qquad z_n = \text{AttnPool}_n(H_{\text{node}}), \qquad z_e = \text{AttnPool}_e(H_{\text{edge}}), \tag{2}$$

where AttnPool_s, AttnPool_n, and AttnPool_e denote separate permutation-invariant attention-based pooling operations. These graph-level latent embeddings are the sole inputs to the decoder, where the recovered spectral coordinates condition the reconstruction of connectivity and attributes.

3.2 Spectral Conditioning Graph Decoder

The decoder reconstructs node-level structure from the graph-level bottleneck through an explicitly spectral representation. The structural latent representation z_s is decoded into node-level spectral coordinates

$$\hat{S} = D_s(z_s). \tag{3}$$

Unlike an unconstrained hidden representation, $\hat{S}$ has an explicit spectral reconstruction target. Its Gram matrix $\hat{S}\hat{S}^\top$ captures pairwise spectral relations between nodes and is used to construct the initial structural channel for the PPGN (Maron et al., 2019).

The node and edge semantic latent representations are decoded using spectral node slots $q_i = [\hat{S}_i \| (\hat{S}\hat{S}^\top)_{ii}]$:

$$\begin{aligned} X_i^{(0)} &= D_n([q_i \| z_n]), \\ E_{ij}^{(0)} &= D_e([q_i \| z_e], [q_j \| z_e]), \end{aligned} \tag{4}$$

where the semantic latents are broadcast over the spectral node slots. The edge decoder combines the features of the two endpoints through a symmetric mapping D_e. These semantic representations and the initial structural channel are fed into the PPGN to jointly reconstruct node categories $\hat{X}$, connectivity $\hat{A}$, and edge types $\hat{E}$. Thus, the predicted spectral representation directly conditions structural and semantic reconstruction, rather than serving only as an auxiliary output.

3.3 Node Matcher

Since the decoded node slots have no predefined correspondence with the input nodes, we define an alignment matrix P that maps predicted outputs to the target node order. Matching costs are indexed by predicted node i and target node j. An initial soft alignment P_0 is obtained by applying Sinkhorn (Cuturi, 2013) to the combined spectral-coordinate and node-category costs:

$$\begin{aligned} C_{ij}^{\text{spec}} &= \|\hat{S}_i - S_j\|_2^2, \qquad C_{ij}^{\text{node}} = \text{CE}(\hat{X}_i, X_j), \\ C_0 &= \alpha_s \bar{C}^{\text{spec}} + \alpha_n \bar{C}^{\text{node}}, \qquad P_0 = \text{Sinkhorn}(-C_0^\top/\tau). \end{aligned} \tag{5}$$

Here, $\tau > 0$ is the temperature controlling the softness of the assignment; smaller values favor sharper correspondences. Sinkhorn alternately normalizes rows and columns in log space, then exponentiates the result to obtain a soft assignment. The spectral term provides structural position cues, while node categories help resolve ambiguities that persist when relying solely on structure.

The initial matching cost compares each node's own spectral coordinates and category. To also compare neighborhood structure, we select r columns from the target spectral coordinates S, and collect them in $Q \in \mathbb{R}^{N \times r}$. Each row of Q provides spectral features for one target node. We use the columns of Q as probes for comparing neighborhoods across the two graphs, with each row providing the probe features of a target node.

Using the preliminary correspondence P_0, we transfer the target features Q to predicted nodes, giving $P_0^\top Q$. Each graph then sums these features over its own neighborhoods:

$$R = AQ, \qquad \hat{R} = \hat{A}P_0^\top Q, \qquad C_{ij}^{\text{probe}} = \|\hat{R}_i - R_j\|_2^2. \tag{6}$$

R_j sums the probe features of target node j's neighbors, while $\hat{R}_i$ aggregates those of predicted node i's neighbors, weighted by the predicted edge probabilities $\hat{A}$. Their discrepancy C_{ij}^{probe} provides a neighborhood cost for refining the correspondence. We add this cost, weighted by α_p, to C_0 to obtain the refined matching cost C. The final alignment P is obtained from C using Sinkhorn during training and the Hungarian (Kuhn, 1955) algorithm during evaluation. This introduces neighborhood consistency without explicitly constructing dense edge-to-edge correspondence costs or an additional learnable module.

3.4 Reconstruction Objectives

We jointly supervise the aligned spectral coordinates and reconstructed graph under the same correspondence P:

$$\begin{aligned} \mathcal{L}_{\text{coord}} &= \frac{\|\operatorname{SignAlign}(P\hat{S}, S) - S\|_F^2}{NK}, \\ \mathcal{L}_{\text{Gram}} &= \operatorname{SmoothL1}(P\hat{S}\hat{S}^\top P^\top, SS^\top), \\ \mathcal{L} &= \lambda_c \mathcal{L}_{\text{coord}} + \lambda_g \mathcal{L}_{\text{Gram}} + \lambda_n \mathcal{L}_{\text{node}} + \lambda_a \mathcal{L}_{\text{adj}} + \lambda_e \mathcal{L}_{\text{edge}}, \end{aligned} \tag{7}$$

Each target spectral column is a scaled Laplacian eigenvector and therefore inherits its sign ambiguity (Lim et al., 2023; Martinkus et al., 2022). We align the predicted and target column signs before computing the coordinate loss. SignAlign independently flips each column of $P\hat{S}$ when its inner product with the corresponding column of S is negative, and otherwise leaves it unchanged. This minimizes the squared error independently for each column, without searching over sign combinations. The Gram term constrains pairwise spectral relations. Both supervise all slots, including the padding nodes. Following the soft reconstruction formulation of Krzakala et al. (2025), graph losses are normalized weighted averages under node weights P and pair weights $P \otimes P$. The terms $\mathcal{L}_{\text{node}}$, $\mathcal{L}_{\text{adj}}$, and $\mathcal{L}_{\text{edge}}$ respectively apply cross-entropy to node categories, binary cross-entropy to edge presence, and cross-entropy to edge types, using the corresponding prediction logits.

4 Experiments

Training datasets. We train G2SAE on three datasets spanning different graph sizes and dataset scales: QM9 (Ramakrishnan et al., 2014) and the PubChem (Kim et al., 2016) subsets PubChem16 and PubChem32. The processed datasets contain 131K, 16.29M, and 94.01M molecules with at most 9, 16, and 32 heavy atoms, respectively. For each dataset, we sample 10,000 molecules for validation and 10,000 for testing, and use the remaining samples for training. Following standard molecular graph representations (Simonovsky & Komodakis, 2018; Krzakala et al., 2025), we omit explicit hydrogen atoms and encode atom and bond types as one-hot node and edge features. We compute the eigenpairs of the unweighted Laplacian $L = D - A$ using a symmetric eigensolver, order them by increasing eigenvalue.

Experimental protocol. Our experiments examine three questions: whether G2SAE improves graph reconstruction over existing graph-level autoencoders, how spectral information affects reconstruction, and how structural and semantic information is distributed across its latent branches.

Table 1: Reconstruction performance of the models. GI Acc. is reported in percent; best available results are bold. N.A. denotes prohibitive training cost for GraphVAE on PubChem32.

	QM9		PubChem16		PubChem32	
Method	GED ↓	GI Acc. ↑	GED ↓	GI Acc. ↑	GED ↓	GI Acc. ↑
GraphVAE	2.28	28.38	3.72	7.82	N.A.	N.A.
PIGVAE	0.38	88.59	1.69	41.00	2.53	24.91
GRALE	0.05	97.91	0.11	93.00	0.78	66.80
G2SAE	**0.03**	**99.23**	**0.05**	**97.44**	**0.57**	**82.07**

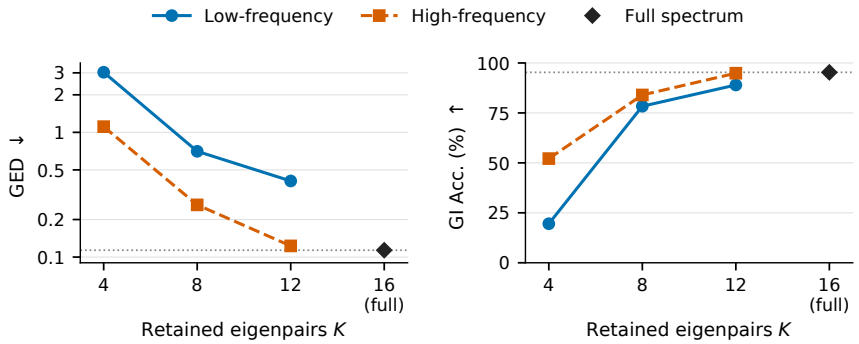


Figure 3: Spectral truncation on PubChem16. Left: GED (log scale); right: GI Acc. Diamonds and dotted lines mark the full-spectrum reference ($K = 16$).

For G2SAE, we use grid-search for hyperparameter tuning and select checkpoints by the lowest validation GED. Appendix A provides the training configurations.

4.1 Reconstruction Performance

We compare our G2SAE with GraphVAE (Simonovsky & Komodakis, 2018), PIGVAE (Winter et al., 2021), and GRALE (Krzakala et al., 2025). We report mean graph edit distance (GED) (Sanfeliu & Fu, 1983) and graph isomorphism accuracy (GI Acc.). We use the same metric equations as GRALE (Krzakala et al., 2025), as detailed in Appendix A.3. GED counts unit-cost node and edge edits under hard node alignment, while GI Acc. is the percentage of reconstructions with zero edits under that alignment.

As shown in Table 1, G2SAE consistently improves both GED and reconstruction accuracy across all three datasets. The difference is most pronounced on PubChem32, where G2SAE improves GI Acc by 15.27 percentage points compared to GRALE. On this larger and more challenging benchmark, the stronger reconstruction performance aligns with our motivation that explicitly reconstructed spectral coordinates provide an informative structural intermediate representation for recovering fine-grained graph connectivity from graph-level latents.

We further investigate the sources of reconstruction error by decomposing GED into node and edge contributions under the same evaluation alignment. By definition, node edits account for node insertions, deletions, and label substitutions, whereas edge edits capture connectivity and edge-label discrepancies. On PubChem16, our model achieves an Edge GED of 0.0424 and a Node GED of 0.0088, with edge errors accounting for approximately 83% of the total GED. A similar concentration of errors on edges is observed for GRALE, consistent with its explicit node-feature supervision and learned node alignment (Krzakala et al., 2025). These observations suggest that recovering structural connectivity remains the principal challenge in graph reconstruction. This motivates our joint spectral and graph reconstruction objectives. By enhancing structural consistency, these objectives address the dominant source of residual error and may also improve graph attribute reconstruction through more coherent relational context.

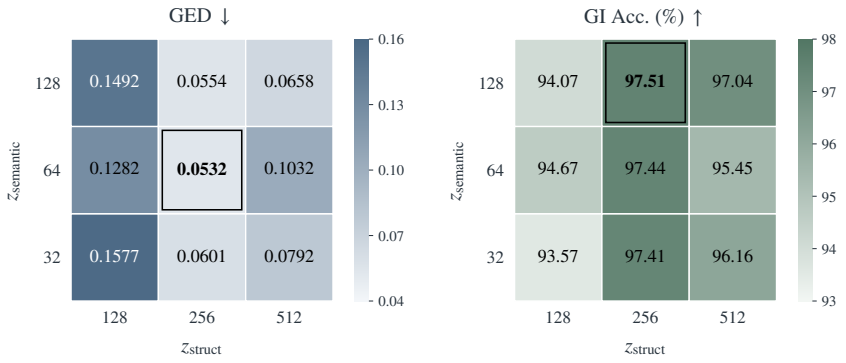


Figure 4: Latent dimensions on PubChem16: GED (left, lighter blue is better) and GI Acc. (right, darker green is better). Outlines mark the best results.

Table 2: Frozen representation probing on QM9 with encoders pretrained on PubChem32. Standardized test MAE (mean ± std) over five dataset splits. Upper: full graph representations; lower: G2SAE branches.

Model / representation	U_0 MAE ↓	ASPL MAE ↓
PIGVAE	0.031 ± 0.001	0.174 ± 0.001
GRALE	0.015 ± 0.001	0.156 ± 0.002
G2SAE (z_{all})	0.010 ± 0.000	0.012 ± 0.001
G2SAE (z_s)	0.578 ± 0.004	0.012 ± 0.002
G2SAE ($z_n \| z_e$)	0.007 ± 0.001	0.098 ± 0.001

4.2 Ablation Experiments

Truncated spectrum. We sample graphs with 16 nodes from PubChem16 so that all samples have the same number of Laplacian eigenpairs before truncation. We compare low- and high-frequency truncations at different values of K for encoding, spectral supervision, and node alignment. As shown in Figure 3, reconstruction improves as more eigenpairs are retained, demonstrating that the spectral representation provides useful structural information for graph recovery. At the same K, high-frequency components consistently outperform low-frequency components, indicating that fine-grained local structural variation is important for recovering graph connectivity.

Latent space dimensions. On PubChem16, we vary the structural and semantic latent dimensions, with the node and edge semantic latents kept at equal dimensions. As shown in Figure 4, increasing latent size does not consistently improve reconstruction. Instead, performance depends on how capacity is distributed between structural and semantic latents, with greater structural capacity generally benefiting graph recovery.

4.3 Latent Space Analysis

Frozen representation probing. We freeze the encoders trained on PubChem32 and fit regression heads on QM9 to predict internal energy (U_0), a molecular property, and average unweighted shortest path length (ASPL), a structural property. We also apply z-score normalization to the targets. Table 2 shows that G2SAE's full representation, $z_{\text{all}} = z_s \| z_n \| z_e$, achieves lower MAE than PIGVAE and GRALE on both tasks. Further analyse the G2SAE results, z_s better predicts ASPL, whereas $z_n \| z_e$ better predicts U_0. This contrasting predictive capability indicates that the various latents tend to retain different information, supporting a functional separation of topological and semantic content.

Molecular editing via latent interpolation. Graph interpolation is related to barycenter and median graph computation (Brogat-Motte et al., 2022; Ferrer et al., 2010; Hlaoui & Wang, 2006). We examine molecular editing by linear latent interpolation, $z_j(t) = (1-t)z_j^A + tz_j^B$. We sample 1,000

Table 3: Quantitative molecular interpolation.

Interpolation	Topology preserved ↑	Connected ↑	Topology GED ↓	Valid edits ↑
All latents	76.53	84.73	0.884	43.90
Fixed z_s	**99.63**	**99.87**	**0.007**	**55.37**

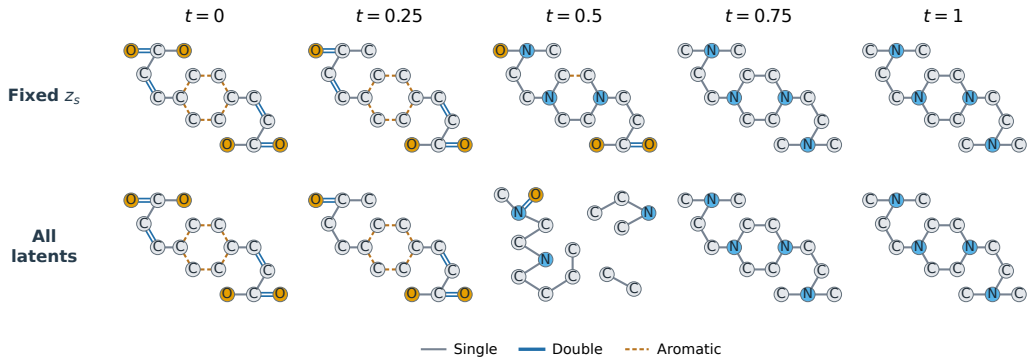


Figure 5: Interpolating graphs from PubChem16. Top: semantic latents interpolated with the structural latent fixed. Bottom: all latents interpolated.

PubChem16 pairs with shared untyped topology. At $t \in \{0.25, 0.5, 0.75\}$, we interpolate z_n, z_e with fixed z_s, or interpolate all three latents. Topology preservation is the fraction of outputs isomorphic to the shared untyped topology, and connectivity is the fraction of connected outputs. Topology GED is the mean exact graph edit distance between untyped output and source graphs. Valid edits are chemically valid, connected outputs that preserve topology and differ from the source. The results in Table 3 show that fixing z_s reduces topological disruption and enables valid, nontrivial molecular changes under the current spectral representation. As shown in Figure 5, interpolating z_n, z_e with $z_s = z_s^A$ permits atom and bond changes while preserving connectivity.

5 Conclusion

We introduce G2SAE, a graph-level autoencoder that uses explicitly supervised spectral coordinates to connect graph reconstruction and node alignment. The model separates structural, node-semantic, and edge-semantic information into graph-level latent representations, and reconstructs node-level spectral structure from the structural latent to guide the recovery of connectivity and attributes. Across QM9, PubChem16, and PubChem32, G2SAE improves both graph edit distance and exact reconstruction accuracy over existing graph-level autoencoders. Further analyses show that spectral information plays a direct role in reconstruction and that the structural and semantic latent branches preferentially capture different graph information. Together, these results show that spectral coordinates can provide a meaningful intermediate representation between graph-level compression and fine-grained graph recovery. Our current experiments focus on molecular graphs with up to 32 heavy atoms. Future work will investigate reconstruction from noisy or corrupted graphs and extend the model toward graph generation, where spectral information may help guide the generation of structurally consistent graphs.

AI USE STATEMENT

We used large language models (LLMs) to assist with drafting and polishing the manuscript, identifying related work, and summarizing research papers, and used AI coding tools to assist in code development. For manuscript preparation, LLMs helped refine wording, improve readability, and clarify the presentation across different sections through sentence rephrasing, grammar checking, and improvements to the overall flow of the text. During code development, LLMs assisted with modifying module code and writing test scripts, following implementation requirements specified by the authors and translating the corresponding testing requirements into executable scripts. The authors reviewed and checked all AI-assisted outputs, including text and code, and take full responsibility for the final manuscript and research results. Both the research ideas and the study design were independently developed by the authors.

ETHICS STATEMENT

This study complies with the ICLR Code of Ethics and involves neither human participants nor animal experiments. All datasets used in this work, including QM9 and the PubChem datasets PubChem16 and PubChem32, were obtained in accordance with the relevant usage guidelines and without compromising privacy. We took care to avoid bias and discriminatory outcomes throughout the research. The study used no personally identifiable information and included no experiments that could raise privacy or security concerns. We remain committed to transparency and integrity throughout the research process.

REPRODUCIBILITY STATEMENT

We have made every effort to ensure the reproducibility of the results reported in this paper. All code and datasets will be made publicly available at the camera-ready stage to support replication and verification. The paper provides a detailed description of the experimental setup, including training procedures, model configurations, and hardware specifications. In addition, the molecular datasets used in this study, namely QM9, PubChem16, and PubChem32, are publicly accessible, ensuring consistency and reproducibility in evaluation. We believe these measures will enable other researchers to reproduce our work and contribute to further progress in the field.

REFERENCES

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, June 2003. ISSN 0899-7667. doi: 10.1162/089976603321780317. URL https://doi.org/10.1162/089976603321780317.

Luc Brogat-Motte, Rémi Flamary, Céline Brouard, Juho Rousu, and Florence d'Alché-Buc. Learning to predict graphs with fused Gromov-Wasserstein barycenters. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2321–2335. PMLR, 2022. URL https://proceedings.mlr.press/v162/brogat-motte22a.html.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/hash/af21d0c97db2e27e13572cbf59eb343d-Abstract.html.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. In *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021. URL https://deep-learning-graphs.bitbucket.io/dlg-aaai21/publications.html.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=wTTjnvGphYj.

Miquel Ferrer, Ernest Valveny, Francesc Serratosa, Kaspar Riesen, and Horst Bunke. Generalized median graph computation by means of graph embedding in vector spaces. *Pattern Recognition*, 43(4):1642–1655, 2010. doi: 10.1016/j.patcog.2009.10.013. URL https://doi.org/10.1016/j.patcog.2009.10.013.

Aditya Grover, Aaron Zweig, and Stefano Ermon. Graphite: Iterative generative modeling of graphs. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2434–2444. PMLR, 2019. URL https://proceedings.mlr.press/v97/grover19a.html.

Hao Guo, Liyuan Deng, Yongkang Dai, Ruohan Wang, Jiahao Li, Yunpeng Bai, and Yilei Shi. BrepVGAE: Variational graph autoencoder with unified latent representation for B-rep. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39230–39238, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Guo_BrepVGAE_Variational_Graph_Autoencoder_with_Unified_Latent_Representation_for_B-rep_CVPR_2026_paper.html.

Judith Hermanns, Anton Tsitsulin, Marina Munkhoeva, Alex Bronstein, Davide Mottin, and Panagiotis Karras. GRASP: Graph alignment through spectral signatures. In *Web and Big Data – APWeb-WAIM 2021*, volume 12858 of *Lecture Notes in Computer Science*, pp. 44–52. Springer, 2021. doi: 10.1007/978-3-030-85896-4_4. URL https://link.springer.com/chapter/10.1007/978-3-030-85896-4_4.

Adel Hlaoui and Shengrui Wang. Median graph computation for graph clustering. *Soft Computing*, 10(1):47–53, 2006. doi: 10.1007/s00500-005-0464-1. URL https://doi.org/10.1007/s00500-005-0464-1.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. GraphMAE: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 594–604. ACM, 2022. doi: 10.1145/3534678.3539321. URL https://doi.org/10.1145/3534678.3539321.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJlWWJSFDH.

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

Wei Ju, Zheng Fang, Yiyang Gu, Zequn Liu, Qingqing Long, Ziyue Qiao, Yifang Qin, Jianhao Shen, Fang Sun, Zhiping Xiao, Junwei Yang, Jingyang Yuan, Yusheng Zhao, Yifan Wang, Xiao Luo, and Ming Zhang. A comprehensive survey on deep graph representation learning. *Neural Networks*, 173:106207, 2024. doi: 10.1016/j.neunet.2024.106207. URL https://doi.org/10.1016/j.neunet.2024.106207.

Sunghwan Kim, Paul A. Thiessen, Evan E. Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A. Shoemaker, Jiyao Wang, Bo Yu, Jian Zhang, and Stephen H. Bryant. PubChem substance and compound databases. *Nucleic Acids Research*, 44(D1):D1202–D1213, 2016. doi: 10.1093/nar/gkv951. URL https://academic.oup.com/nar/article/44/D1/D1202/2503131.

Thomas N. Kipf and Max Welling. Variational graph auto-encoders. NIPS Workshop on Bayesian Deep Learning, 2016. URL https://bayesiandeeplearning.org/2016/.

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking Graph Transformers with Spectral Attention. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21618–21629. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/b4fd1d2cb085390fbbadae65e07876a7-Abstract.html.

Paul Krzakala, Gabriel Melo, Charlotte Laclau, Florence d'Alché-Buc, and Rémi Flamary. The quest for the GRAph Level autoEncoder (GRALE). In *Advances in Neural Information Processing Systems*, volume 38, pp. 58194–58229. Curran Associates, Inc., 2025. doi: 10.52202/085713-1950. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/547108084f0c2af39b956f8eadb75d1b-Abstract-Conference.html.

H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955. doi: 10.1002/nav.3800020109. URL https://doi.org/10.1002/nav.3800020109.

Derek Lim, Joshua David Robinson, Lingxiao Zhao, Tess E. Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and basis invariant networks for spectral graph representation learning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Q-UHqMorzil.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bb04af0f7ecaee4aae62035497da1387-Abstract.html.

Karolis Martinkus, Andreas Loukas, Nathanaël Perraudin, and Roger Wattenhofer. SPECTRE: Spectral conditioning helps to overcome the expressivity limits of one-shot graph generators. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15159–15179. PMLR, July 2022. URL https://proceedings.mlr.press/v162/martinkus22a.html.

Giorgia Minello, Alessandro Bicciato, Luca Rossi, Andrea Torsello, and Luca Cosmo. Generating graphs via spectral diffusion. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=AAXBfJNHDt.

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. doi: 10.1038/sdata.2014.22. URL https://www.nature.com/articles/sdata201422.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, volume 35, pp. 14501–14515. Curran Associates, Inc., 2022. doi: 10.52202/068431-1054. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/5d4834a159f1547b267a05a4e2b7cf5e-Abstract-Conference.html.

Alberto Sanfeliu and King-Sun Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(3):353–362, May 1983. ISSN 2168-2909. doi: 10.1109/TSMC.1983.6313167. URL https://doi.org/10.1109/TSMC.1983.6313167.

David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains. *IEEE Signal Processing Magazine*, 30(3):83–98, May 2013. ISSN 1053-5888. doi: 10.1109/MSP.2012.2235192. URL https://doi.org/10.1109/MSP.2012.2235192.

Martin Simonovsky and Nikos Komodakis. GraphVAE: Towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning – ICANN 2018*, volume 11139 of *Lecture Notes in Computer Science*, pp. 412–422. Springer, 2018. doi: 10.1007/978-3-030-01418-6_41. URL https://doi.org/10.1007/978-3-030-01418-6_41.

Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1lfF2NYvH.

Runzhong Wang, Junchi Yan, and Xiaokang Yang. Learning combinatorial embedding networks for deep graph matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3056–3065. IEEE, October 2019. ISBN 978-1-7281-4803-8. doi: 10.1109/ICCV.2019.00315. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Wang_Learning_Combinatorial_Embedding_Networks_for_Deep_Graph_Matching_ICCV_2019_paper.html.

Robin Winter, Frank Noé, and Djork-Arné Clevert. Permutation-invariant variational autoencoder for graph-level representation learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 9559–9573. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/4f3d7d38d24b740c95da2b03dc3a2333-Abstract.html.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Carl Yang, Peiye Zhuang, Wenhan Shi, Alan Luu, and Pan Li. Conditional structure generation through graph variational generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/e57c6b956a6521b28495f2886ca0977a-Abstract.html.

Muhan Zhang, Shali Jiang, Zhicheng Cui, Roman Garnett, and Yixin Chen. D-VAE: A variational autoencoder for directed acyclic graphs. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/e205ee2a5de471a70c1fd1b46033a75f-Abstract.html.

Ziwei Zhang, Peng Cui, and Wenwu Zhu. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):249–270, 2022. doi: 10.1109/TKDE.2020.2981333. URL https://doi.org/10.1109/TKDE.2020.2981333.

A Experimental settings and implementation details

A.1 Datasets and graph preprocessing

Molecular graphs provide discrete atom and bond labels together with connectivity, allowing us to assess how well G2SAE preserves both structural and semantic information. We use QM9 (Ramakrishnan et al., 2014) and two size-restricted subsets of PubChem (Kim et al., 2016), PubChem16 and PubChem32, covering different graph sizes and training-data scales.

Following the molecular graph representation of Krzakala et al. (2025), we represent molecules as undirected graphs with hydrogen atoms omitted, using atom types as node labels and bond types as edge labels. For PubChem, we convert SMILES strings into graphs with RDKit and discard molecules whose conversion fails or whose graph size exceeds the corresponding dataset limit. For QM9, we use the graph data provided by PyTorch Geometric and remove hydrogen nodes and their incident edges. Atom and bond labels are encoded as one-hot features.

A.2 Model configurations

Table 5 reports the configurations used for the G2SAE reconstruction results. Structural, node-semantic, and edge-semantic latent dimensions are denoted by d_s, d_n, and d_e. Each model was trained on one NVIDIA RTX 4090.

Table 4: Processed datasets for graph reconstruction. Node categories include padding; edge categories include the no-edge class.

Property	QM9	PubChem16	PubChem32
Training graphs	110,828	16,267,483	93,989,120
Validation graphs	10,000	10,000	10,000
Test graphs	10,000	10,000	10,000
Total graphs	130,828	16,287,483	94,009,120
Maximum heavy atoms	9	16	32
Node categories	5	33	33
Edge categories	4	6	6

Table 5: G2SAE model training settings. Width/depth denotes hidden dimension and layer count.

Setting	QM9	PubChem16	PubChem32
Retained eigenpairs K	9	16	32
Spec PE hidden dimension	32	32	64
Topology GIN width/depth	$256/6$	$256/2$	$512/3$
Semantic GINE width/depth	$64/2$	$64/2$	$64/2$
Latent dimensions (d_s, d_n, d_e)	$(256, 96, 96)$	$(256, 64, 64)$	$(512, 96, 96)$
Semantic initializer width/depth	$64/2$	$64/2$	$128/3$
PPGN width/depth	$256/6$	$256/6$	$256/6$
Model parameters (M)	5.47	4.48	8.41
Spectral coordinate loss weight λ_c	1	100	100
Spectral Gram loss weight λ_g	0.5	0.1	0.1
Node loss λ_n	5	0.5	0.1
Adjacency loss λ_a	20	0.5	1
Edge-type loss λ_e	0.5	0.1	0.1
Spectral/node matching weights	$0.8/0.2$	$0.8/0.2$	$0.8/0.2$
Spectral probes	5	8	8
Final probe weight α_p	0.03	0.1	0.1
Batch size	128	64	64
Graph draws per epoch	110,720	256,000	256,000
Maximum optimization steps	500,000	2,000,000	2,000,000
Mean throughput (steps/s)	≈ 18.81	≈ 17.36	≈ 7.47

A.3 Evaluation metrics

Following GRALE (Krzakala et al., 2025), we assign unit cost to node and edge insertions, deletions, and label substitutions. For metric computation, write the target graph and its discrete reconstruction as $G = (F, C)$ and $\hat{G} = (\hat{F}, \hat{C})$, both padded to N node slots. Here, F contains node labels, including the padding category, and C contains pairwise edge labels, including the no-edge category. For a permutation matrix P, define

$$\begin{aligned} P[\hat{G}] &= \left(P\hat{F}, P\hat{C}P^\top\right), \\ \mathcal{L}_{\text{align}}(G, \hat{G}) &= \sum_{i=1}^{N} \mathbb{I}[F_i \neq \hat{F}_i] + \sum_{1 \leq i < j \leq N} \mathbb{I}[C_{ij} \neq \hat{C}_{ij}], \end{aligned} \tag{8}$$

where $\mathbb{I}[\cdot]$ is the indicator function. Padding and no-edge labels allow insertions and deletions to be counted within the same representation. The second sum counts each unordered node pair once and excludes self-loops.

Table 6: Frozen representation probing on QM9 U_0 with encoders pretrained on PubChem16 under different spectral truncation settings.

Checkpoint	z_s	$z_n \| z_e$	z_{all}
Lowest $K = 4$	0.581 ± 0.005	0.013 ± 0.001	0.011 ± 0.001
Lowest $K = 8$	0.581 ± 0.003	0.013 ± 0.001	0.013 ± 0.000
Lowest $K = 12$	0.579 ± 0.005	0.010 ± 0.000	0.010 ± 0.000
Highest $K = 4$	0.582 ± 0.004	0.008 ± 0.001	0.021 ± 0.008
Highest $K = 8$	0.580 ± 0.005	0.009 ± 0.001	0.011 ± 0.001
Highest $K = 12$	0.578 ± 0.004	0.010 ± 0.001	0.010 ± 0.001

Table 7: Frozen representation probing on QM9 ASPL with encoders pretrained on PubChem16 under different spectral truncation settings.

Checkpoint	z_s	$z_n \| z_e$	z_{all}
Lowest $K = 4$	0.008 ± 0.002	0.265 ± 0.003	0.008 ± 0.001
Lowest $K = 8$	0.009 ± 0.001	0.238 ± 0.002	0.010 ± 0.001
Lowest $K = 12$	0.008 ± 0.001	0.239 ± 0.003	0.009 ± 0.002
Highest $K = 4$	0.013 ± 0.002	0.173 ± 0.002	0.011 ± 0.000
Highest $K = 8$	0.013 ± 0.001	0.244 ± 0.003	0.012 ± 0.001
Highest $K = 12$	0.007 ± 0.002	0.243 ± 0.002	0.009 ± 0.002

At evaluation, let P_{H} be the hard Hungarian assignment returned by our matching criterion. We compute the edit count under this correspondence as

$$\mathrm{GED}_{\mathrm{H}}(G, \hat{G}) = \mathcal{L}_{\text{align}}\Big(G, P_{\mathrm{H}}[\hat{G}]\Big). \tag{9}$$

Over an evaluation set $\mathcal{D}$ of target–reconstruction pairs, we report the mean hard-aligned edit count as GED and define GI Acc. as the percentage of reconstructions with zero hard-aligned edits:

$$\begin{aligned} \overline{\mathrm{GED}}_{\mathrm{H}} &= \frac{1}{|\mathcal{D}|} \sum_{(G,\hat{G})\in\mathcal{D}} \mathrm{GED}_{\mathrm{H}}(G, \hat{G}), \\ \mathrm{GIAcc}_{\mathrm{H}} &= \frac{100}{|\mathcal{D}|} \sum_{(G,\hat{G})\in\mathcal{D}} \mathbb{I}\Big[\mathrm{GED}_{\mathrm{H}}(G, \hat{G}) = 0\Big]. \end{aligned} \tag{10}$$

Lower mean GED and higher GI Acc. indicate better reconstruction. Node GED and Edge GED are obtained by separately averaging the node and edge terms in Eq. 8 under the same hard alignment.

A.4 Downstream task with truncated spectrum

In Tables 6 and 7 evaluate the performance of frozen GS2AE encoders trained on truncated spectral information on downstream regression tasks. Across all spectral truncation settings, the structural branch predicts ASPL more accurately than the semantic branches, whereas the semantic branches better predict U_0, indicating consistent differences in the information accessible from these representations. At smaller truncation ranks ($K = 4, 8$), encoders pretrained with the lowest-frequency modes yield lower ASPL MAE for both zs and $z\mathrm{all}$ than their highest-frequency counterparts. These results highlight the computational complexity advantage of our encoder (quadratic in the number of graph nodes when the spectrum is truncated) over that of GRALE (cubic in the number of graph nodes).