# POLYGRAPHSCORE: A CLASSIFIER-BASED METRIC FOR EVALUATING GRAPH GENERATIVE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Existing methods for evaluating graph generative models primarily rely on Maximum Mean Discrepancy (MMD) metrics based on graph descriptors. While these metrics can rank generative models, they do not provide an absolute measure of performance. Their values are also highly sensitive to extrinsic parameters, namely kernel and descriptor parametrization, making them incomparable across different graph descriptors. We introduce `PolyGraphScore` (PGS), a new evaluation framework that addresses these limitations. It approximates the Jensen-Shannon (JS) distance of graph distributions by fitting binary classifiers to distinguish between real and generated graphs, featurized by these descriptors. The data log-likelihood of these classifiers approximates a variational lower bound on the JS distance between the two distributions. Resulting scores are constrained to the unit interval $[0, 1]$ and are comparable across different graph descriptors. We further derive a theoretically grounded summary score that combines these individual metrics to provide a maximally tight lower bound on the distance for the given descriptors. Thorough experiments demonstrate that PGS provides a more robust and insightful evaluation compared to MMD metrics.

## 1 INTRODUCTION

Graph generative models (GGMs) are seeing wider adoption across scientific domains, from retrosynthesis (Somnath et al., 2021) and social network modeling (Bojchevski et al., 2018) to the discovery of novel drugs and materials (Liu et al., 2024; Kelvinius et al., 2025). However, progress in this field is increasingly bottlenecked by the lack of robust methods for evaluating generated graphs (Thompson et al., 2022; O'Bray et al., 2022).

This evaluation challenge is not unique to graphs. In image generation, the community has largely converged on pretrained embeddings paired with distribution distances, such as Inception-v3 coupled with Fréchet distance yielding the widely used Fréchet Inception distance (FID) (Heusel et al., 2017), or DinoV2 and density estimation producing the Feature Likelihood Divergence (FLD) (Jiralerspong et al., 2023). While these approaches provide standardized metrics adapted to other fields such as materials (Kelvinius et al., 2025), video (Unterthiner et al., 2019), and audio (Kilgour et al., 2018), limitations remain (Barratt & Sharma, 2018). As an alternative, classifier two-sample tests (C2STs) (Lopez-Paz & Oquab, 2017) recasts evaluation as a supervised classification task, turning classifier performance into evaluation metrics. To date, the applicability of these approaches to graph-structured data has not yet been explored.

The *de facto* standard for evaluating GGMs is to compute the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) between distributions of hand-crafted graph descriptors (e.g., degrees, Laplacian spectra, etc.) on a small set of synthetic and real-world graphs (You et al., 2018). While convenient, this approach has critical inherent limitations: (i) MMD estimates lack an *intrinsic scale*, meaning that a single reported value without context does not provide an absolute notion of the goodness of fit of the generative model; (ii) rankings are *sensitive to descriptor and kernel choice* (O'Bray et al., 2022), with no way of obtaining a single ranking across descriptors for consistent and systematic model comparison; and (iii) in the small-sample regimes common to current GGM benchmarks, MMD estimates suffer from *high bias and variance* (Krimmel et al., 2025)

We introduce `PolyGraphScore` (PGS), a novel evaluation framework that estimates the Jensen-Shannon distance (JSD) (Endres & Schindelin, 2003) between true and generated graph distributions using *probabilistic classification* instead of kernel-based distances. A discriminator is trained to distinguish real from generated graphs using standard graph descriptors, where the classifier's data log-likelihood provides a lower bound on the JSD. This yields scores in $[0, 1]$ that are directly *comparable across descriptors*. Taking the maximum over descriptors gives the tightest available bound while identifying the most informative descriptor.

Our formulation of PGS uses TabPFN (Hollmann et al., 2025), a fast, hyperparameter-free discriminator, making it robust and simple to use. Empirically, we show that PGS monotonically tracks synthetic data perturbations, strongly correlates with model training progress, and accurately captures generated graph quality. It also produces robust rankings across representative GGMs. Table 1 summarizes the advantages of PGS over MMD.

Table 1: Comparison of Maximum Mean Discrepancy and `PolyGraphScore`.

| Property | MMD | PGS |
|---|---|---|
| Range | $[0, \infty)$ | $[0, 1]$ |
| Intrinsic Scale | ✗ | ✓ |
| Descriptor Comparison | ✗ | ✓ |
| Single Ranking | ✗ | ✓ |

Our work makes four primary contributions:

- **A rigorous reassessment of MMD for GGM evaluation.** We empirically show that standard MMD estimators are plagued by high bias and variance at typical benchmark sizes (20-40 graphs), leading to unreliable model rankings, and we provide actionable remedies.
- **`PolyGraphScore` (PGS): an estimate of the JSD distance between distributions.** We propose a method to derive interpretable evaluation scores by approximating variational lower bounds on the JSD via probabilistic discrimination on graph descriptors.
- **A comprehensive empirical validation.** We show that PGS tracks data perturbations monotonically and correlates strongly with training dynamics of state-of-the-art models. We also provide comprehensive PGS-based benchmark results across synthetic and real-world graphs, including molecules.
- **An open-source library to advance GGM evaluation.** We release the `PolyGraph` library, including implementations of PGS, MMD estimators, and new, larger benchmark datasets (SBM-L, Lobster-L, Planar-L), to facilitate more robust and reproducible future research.

## 2 Related Work

We present here related work on the evaluation of graph generative models and classifier-based evaluation for general generative models.

**Evaluation of Graph Generative Models.** The evaluation of GGMs has largely been shaped by methods based on the MMD (Gretton et al., 2012). You et al. (2018) first proposed computing the MMD between generated and real graph distributions using a Wasserstein Gaussian kernel on a set of graph descriptors, including degree histograms, clustering coefficients, and orbit counts. To reduce the computational cost of this method, Liao et al. (2019) introduced a simpler kernel formulation using a Gaussian kernel with the squared total variation distance, which gained widespread adoption (Martinkus et al., 2022; Vignac et al., 2023; Chen et al., 2025). However, this simplified kernel was shown to be indefinite and highly sensitive to hyperparameter choices (O'Bray et al., 2022). Subsequent work has focused on correcting these flaws, either by modifying the kernel to ensure it is positive definite (O'Bray et al., 2022) or by employing standard RBF kernels with automated hyperparameter tuning (Thompson et al., 2022; Sriperumbudur et al., 2009). A parallel research effort has concentrated on identifying more expressive graph descriptors for use within the MMD framework. The initial set of statistics was augmented with the graph Laplacian spectrum by Liao et al. (2019), and more recently, graph neural networks (GNNs) have been used as powerful graph featurizers (Thompson et al., 2022; Shirzad et al., 2022). Despite these advances, a key limitation remains: *since MMD has no inherent scale, it is difficult to assess whether newly proposed descriptors are suited for discriminating between real and generated graphs.* PGS, however, is comparable across descriptors and thus explicitly quantifies their discriminative power.

Departing from MMD, other evaluation paradigms have been proposed. Southern et al. (2023) used tools from topological data analysis, featurizing graphs via persistent homology and comparing distributions based on their average persistence landscapes. In a different direction, Martinkus et al. (2022) introduced synthetic benchmark datasets (Planar and SBM) that allow for judging the structural validity of individual graph samples–such as planarity. The small size of the synthetic datasets and the resulting variance in MMD estimates were criticized by Krimmel et al. (2025). *We expand on these observations and propose concrete techniques for quantifying the uncertainty in GGM evaluation metrics, addressing a critical need for more reliable and reproducible evaluations.*

**Classifier-Based Evaluation.** One relevant family of metrics used for generative model evaluation is derived from the classifier two-sample test (C2ST) (Lopez-Paz & Oquab, 2017). This work proposes to discriminate generated from reference samples via binary classification and repurpose the resulting accuracy as a measure for the separability of the generated and reference distributions. By extension, it assesses the quality of the generative model.

The MMD can also be viewed through this lens, as it corresponds to the optimal linear risk of a kernel classifier (Sriperumbudur et al., 2009; Gretton & Jitkrittum, 2016). Generative adversarial networks (GANs) (Goodfellow et al., 2014; Li et al., 2015; Bińkowski et al., 2018; Arjovsky et al., 2017) also leverage a classifier's output, not just for training but also for evaluation, where classifier-based divergences (including MMD) have been shown to correlate well with the perceptual quality of generated images (Im et al., 2018).

Despite the success of these methods in other domains, their application to graph generation has been limited. While some work has used fixed multi-class classifiers on generated graphs to measure performance (Liu et al., 2019), classifiers that discriminate between real and generated graphs have not been explored beyond the MMD framework. *Our work addresses this gap, proposing a novel classifier-based evaluation framework for GGMs that provides scores that are (i) absolute, (ii) comparable across different graph descriptors, and (iii) capable of estimating lower bounds on certain probability metrics.*

## 3 Preliminaries

In this section, we review two divergences, MMD and the Jensen-Shannon (JS) divergence, from a unified variational perspective: the optimal performance of a discriminator tasked with distinguishing between two distributions. We first discuss MMD, interpreting it as the linear risk of a classifier in a reproducing kernel Hilbert space (RKHS) (Sriperumbudur et al., 2009). We highlight its limitations in the context of graph generation, primarily its lack of an absolute scale, which motivates our subsequent review of the JS distance as a foundation for more interpretable, classifier-based evaluation metrics such as the `PolyGraphScore`.

### 3.1 MMD and its Interpretation as Classification Risk

Given two probability distributions $P$ and $Q$ over a space $\mathcal{X}$ (in our case, the space of graphs) and a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, the squared MMD is defined as:

$$\mathrm{MMD}^2(P, Q, k) := \mathbb{E}_{x,x' \sim P}[k(x, x')] - 2\mathbb{E}_{x\sim P, y\sim Q}[k(x, y)] + \mathbb{E}_{y,y' \sim Q}[k(y, y')]. \tag{1}$$

The MMD can be expressed as the distance between the mean embeddings of $P$ and $Q$ in the RKHS $\mathcal{H}$ induced by $k$. This framing leads to a variational formulation where the MMD is precisely the optimal linear classification risk achievable by a discriminator in the unit ball of $\mathcal{H}$ (Sriperumbudur et al., 2009). We refer to Appendix D for a detailed derivation.

**Limitations.** A fundamental limitation of MMD for model evaluation is its lack of an absolute scale (O'Bray et al., 2022). The MMD value is sensitive to the choice of kernel and the scaling of input features. For instance, using a linear kernel, simply scaling the input features by a scalar factor will scale the resulting MMD by the same factor. This makes it impossible to compare MMD scores across different graph descriptors. While MMD can rank models relative to a baseline for a fixed descriptor, it provides no absolute measure of performance.

To overcome this, we turn to metrics that possess a fixed intrinsic scale, making them comparable across different graph descriptors. This leads us to the Jensen-Shannon divergence and, more generally, to the family of $f$-divergences.

### 3.2 Variational Estimation of the Jensen-Shannon Distance

The Jensen-Shannon (JS) divergence is a symmetrized version of the Kullback-Leibler (KL) divergence: $\frac{1}{2}(D_{\mathrm{KL}}(P \,\|\, M) + D_{\mathrm{KL}}(Q \,\|\, M))$ with $M := \frac{1}{2}(P + Q)$ being the mixture of $P$ and $Q$. It is

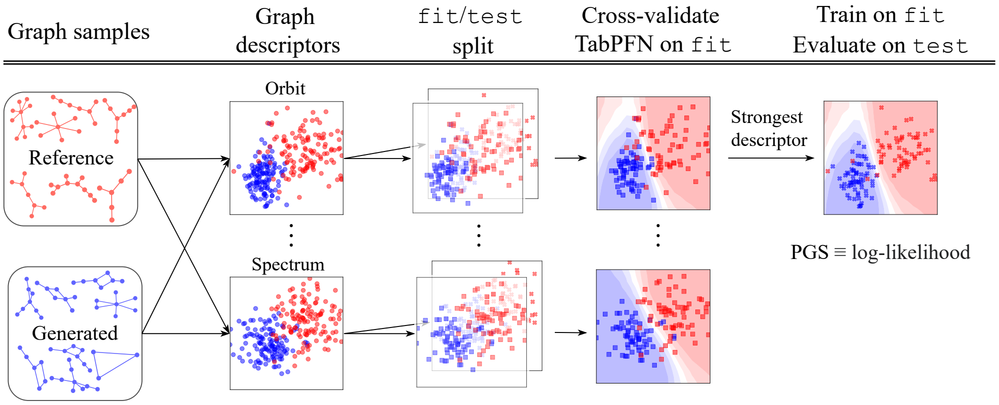


Figure 1: Computation of the PGS metric. TabPFN is trained to discriminate between generated and reference graphs based on different vectorial descriptions. The most expressive descriptor (here: orbit) is used to derive the final PGS, yielding a maximally tight lower bound on the Jensen-Shannon (JS) distance between the generated and reference graph distributions.

constrained to the unit interval $[0, 1]$ and, in contrast to MMD, is independent of extrinsic parameters such as kernel choice. As extensively leveraged in GANs (Goodfellow et al., 2014), the JS divergence admits (under mild conditions) a variational formulation as the maximal data log-likelihood (up to constants) achievable by a binary classifier $D$ distinguishing between samples from $P$ and $Q$:

$$D_{\text{JS}}(P \parallel Q) = \sup_{D:\mathcal{X}\to[0,1]} \frac{1}{2}\mathbb{E}_{x\sim P}[\log_2 D(x)] + \frac{1}{2}\mathbb{E}_{x\sim Q}[\log_2(1 - D(x))] + 1. \tag{2}$$

Importantly, the log-likelihood of *any* classifier provides a valid lower bound on the JS divergence and the bound is tightened by fitting a classifier via maximum likelihood methods. While the JS divergence is not a metric, its square root (termed the JS distance) is (Endres & Schindelin, 2003).

The JS divergence belongs to the larger family of $f$-divergences. As shown by Nguyen et al. (2010), any $f$-divergence admits a variational formulation similar to Eqn. (2). In Appendix E, we investigate the total variation (TV) distance as a possible alternative to the JS distance. Instead of log-likelihood, we show that the variational objective of the TV distance is given by the classifier's *informedness*.

## 4 PolyGraphScore: Variational Estimates of the JS Distance

Building on the variational view of divergences, we introduce PolyGraphScore (PGS), a framework for evaluating GGMs. PGS estimates the JS distance between a distribution of reference graphs and a distribution of generated graphs. The core idea is to reframe the divergence estimation as a classification task: we featurize graphs using a variety of established graph descriptors and measure how well a powerful, non-parametric classifier can distinguish between the two sets. The resulting classifier performance, measured in terms of log-likelihood, serves as a tight, empirical lower bound on the true JS divergence between the underlying graph distributions. Fig. 1 shows this procedure.

Our method proceeds in two main stages. First, we detail how to estimate a lower bound on the divergence using a *single* graph descriptor in Section 4.1. Second, we describe in Section 4.2 how to systematically combine *multiple* descriptors from a larger set to compute the final PGS, which represents the tightest lower bound from the given descriptors. We provide pseudocode in Appendix B.

### 4.1 Estimating the JS Distance with a Single Descriptor

Given a multiset of reference graphs $P_{\text{ref}}$ and generated graphs $Q_{\text{gen}}$, along with a single graph descriptor $d : \mathcal{X} \to \mathbb{R}^n$, we estimate the divergence of $P_{\text{ref}}$ and $Q_{\text{gen}}$ via featurization by $d$.

To prevent overfitting, where a classifier might perfectly memorize the training data and thus overestimate the true divergence, we randomly partition both $P_{\text{ref}}$ and $Q_{\text{gen}}$ into disjoint fit and test sets of equal size. Our goal is to approximate the supremum in Eqn. (2) by training a discriminator exclusively on the fit set, and computing the final divergence estimate on the held-out test set.

**Discriminator Choice.** An appropriate discriminator for this task must satisfy three criteria:

1. **Probabilistic:** It must output class probabilities to estimate the JS divergence via its log-likelihood objective.
2. **Efficient:** It must be fast to train, enabling rapid evaluation across many descriptors.
3. **Hyperparameter-Free:** It should be robust and require no manual tuning to ensure fair and reproducible comparisons.

These requirements rule out the training of deep neural networks with gradient descent, because it is computationally expensive and requires hyperparameter tuning. It also rules out non-probabilistic models such as decision trees and SVMs. As a result, we choose TabPFN (Hollmann et al., 2025) in this work. TabPFN is a transformer-based model that approximates Bayesian inference over a large space of simple models consisting of Bayesian neural networks and structural causal models. It is fast (see Table 15), requires no hyperparameter tuning, and has proven to be a powerful classifier for tabular data, making it an ideal choice for our framework since our classifier operates on graph descriptors. In Appendix J, we investigate logistic regression as an alternative choice and show that TabPFN yields tighter bounds in practice. In Appendix R we show that kernel logistic regression also fits naturally into the PGS framework, allowing for the use of, *e.g.*, graph kernels (Borgwardt & Kriegel, 2005; Shervashidze et al., 2011; Grauman & Darrell, 2007). However, similar to logistic regression, we found that those kernel logistic regression-based PGS scores yielded looser bounds in practice, and elected to proceed with TabPFN.

**Estimation Procedure.** With a discriminator selected, we first apply the descriptor $d : \mathcal{X} \to \mathbb{R}^n$ to the graphs in the fit set to create vectorial features. We then train the binary classifier on these features using TabPFN. We apply the descriptor to the test set and use the trained classifier to evaluate the data log-likelihood, providing an approximate lower bound of the JS divergence. Finally, we take the square root to estimate the JS *distance*.

### 4.2 Descriptor Selection for the Tightest Bound

A single graph descriptor captures only one specific aspect of graph structure. To obtain a more comprehensive evaluation, we consider a collection of $K$ distinct descriptors $\{d_1, \ldots, d_K\}$. The goal is to identify the single descriptor that most effectively distinguishes between the reference and generated graphs, as this descriptor will yield the tightest lower bound on the true JS distance. This descriptor selection process must be performed carefully to avoid data leakage from the test set, which would invalidate our final estimate. We therefore perform selection using only the fit data via cross-validation.

**Cross-Validation on the Fit Set.** For each descriptor $d_k : \mathcal{X} \to \mathbb{R}^n$, we estimate its ability to separate the distributions by performing 4-fold stratified cross-validation on the $(P_{\text{ref}}^{\text{fit}}, Q_{\text{gen}}^{\text{fit}})$ data. In each fold, three-quarters of the data are used for training a discriminator, and the remaining quarter is used for validation. The average validation score across the four folds provides a robust estimate of the lower bound achievable by that descriptor.

**The PolyGraphScore.** After performing cross-validation for all $K$ descriptors, we select the descriptor $d^\star$ that yielded the highest average score. This is the descriptor that is empirically the most informative. Finally, we train a new discriminator for $d^\star$ on the *entire* fit set and evaluate it on the held-out test set. The resulting score is the PolyGraphScore (PGS). This procedure ensures that the descriptor selection and final evaluation are performed on separate data, yielding a principled and tight estimate of the divergence between the graph distributions.

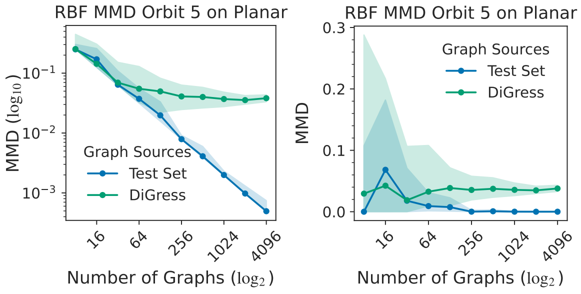
## 5 Experiments

We empirically validate PGS through a series of experiments designed to test its robustness, sensitivity, and practical utility against standard MMD-based metrics for evaluating graph generative models. Our investigation consists of four stages:

- First, Section 5.1 shows that MMD evaluations suffer from *substantial bias and variance on current datasets*, motivating the use of larger datasets, unbiased estimators, and subsampling to assess estimate stability.
- In Section 5.2, we show that *PGS correlates well with controlled perturbations* applied to synthetic datasets, showing the power of JSD to distinguish samples from different distributions.
- In a realistic use case for a state-of-the-art diffusion model (Section 5.3), we show that *PGS reliably tracks training progress and performance gains* when increasing the number of denoising steps. Our results indicate that PGS captures model quality more reliably than MMD metrics.
- Finally, in Section 5.4 we leverage PGS to conduct a *comprehensive benchmark* of several representative GGMs.

Unless otherwise stated, all PGS scores are based on the Jensen-Shannon (JS) distance estimated using TabPFN as the discriminator. Following previous works (You et al., 2018; Liao et al., 2019; Thompson et al., 2022), we use degree histograms (abbreviated as Degree/Deg. in our tables and figures), clustering coefficient histograms (Clustering/Clust.), the Laplacian spectrum (Spectral/Spec.), orbit counts (Orbit), and GIN embeddings (GIN) as descriptors. For molecular graphs, we use domain-specific descriptors based on topological properties, physico-chemical parameters, and learned representations. We refer to Appendix C for further details.

### 5.1 High Bias and Variance Plague MMD-based GGM Benchmarks



(a) Biased MMD.

(b) Unbiased MMD.

Figure 2: Examples of MMD estimates that suffer from high bias (left) and variance (right).

The evaluation of GGMs is predominantly conducted on synthetic, procedurally generated datasets, including lobster graphs, stochastic block models (SBMs), and planar graphs, which permit the generation of arbitrarily large numbers of samples. Krimmel et al. (2025) first raised the issue that MMD values computed on such datasets can exhibit considerable variance, thereby casting doubt on the robustness of model rankings derived from these metrics. In order to more rigorously characterize this phenomenon, we exploited the procedural nature of these datasets to systematically vary the subsample sizes used in MMD. The MMD shown here is obtained with the radial basis function (RBF) kernel; more details are given in Appendix G.

In the regime of commonly used synthetic graph benchmarks (between 20 and 40 test graphs, c.f. Appendix P), bias dominates the MMD values (Figure 2a, in log scale for clarity). Even when using the unbiased MMD estimator[1], the variance across subsamples remains large enough to make model comparisons at these sample sizes unreliable (Figure 2b). Figure 2 illustrates these issues for DiGress-generated samples for planar graphs described with orbit counts, but extensive experiments in Appendix G show that they persist across all combinations of models, descriptors, and datasets.

This finding yields three actionable insights. First, prefer *unbiased MMD* estimates, as bias depends heavily on sample size. Second, akin to Krimmel et al. (2025), use *larger sample sizes* to reduce estimator variance; we propose SBM-L, PLANAR-L, and LOBSTER-L for this purpose (with more details in Appendix M)[2]. Third, report the *variance* of MMD across subsamples to quantify the stability of the estimates. To assess the effect of dataset size on PGS, we conducted analogous experiments in Appendix L, which show that its mean and variance stabilize beyond subsample sizes of about 256. This is particularly relevant because TabPFN's discriminative power may depend on sample size.

[1] Our MMD estimates are not unbiased, as we take the maximum MMD value over a set of kernel bandwidths, but we do use the unbiased MMD estimate without diagonals, see Eq. 3 in Gretton et al. (2012).

[2] AutoGraph reaches similar VUN scores with markedly lower loss in SBM-L than on the original SBM dataset (see Appendix N), showing reduced overfitting, which is underexplored in GGMs (Vignac et al., 2023).

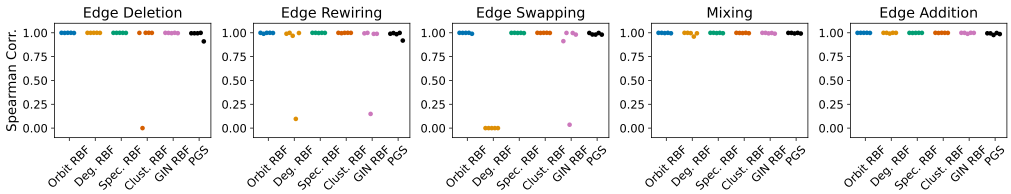


Figure 3: Spearman correlation of MMDs and PGS with magnitude of perturbation.

### 5.2 PolyGraphScore Tracks Synthetic Data Perturbations

To validate PGS as a reliable metric, we verify its ability to correlate with the magnitude of perturbations applied to graph datasets, a standard procedure for evaluating graph metrics (O'Bray et al., 2022; Thompson et al., 2022). Our experiments demonstrate that PGS effectively tracks these changes, performing on par with MMDs.

**Experimental Setup.** We conduct our experiments on five datasets: Protein contact graphs (Dobson & Doig, 2003), ego nets extracted from Citeseer (Sen et al., 2008), and three procedural datasets (Planar, SBM, Lobster). Each procedural dataset contains 4096 samples, while the proteins dataset contains 918 samples, and the ego dataset contains 757 samples. Dataset details are in Appendix P.

To simulate data corruption, we apply five distinct perturbation types, four of which are adapted from previous studies (O'Bray et al., 2022; Thompson et al., 2022). Each perturbation modifies the graph structure (or dataset) in a controlled manner. Edge deletion/addition removes or adds a specified number of edges selected at random. Edge rewiring replaces one of the incident vertices of some edges with a randomly selected vertex. Mixing operates on the dataset level by replacing a fraction of the graphs within a dataset with new samples from an Erdős–Rényi model. Finally, we propose a novel perturbation type which we term "edge swapping". Edge swapping selects pairs of edges and swaps two of their incident vertices. This transformation preserves the vertex degrees, making it a more challenging perturbation for some metrics to detect.

**Perturbation Experiments.** Our core experiment involves splitting each dataset into two equal subsets: one serves as a fixed reference distribution, and the other is subjected to the perturbations. We then measure the distance between the reference and the perturbed subset using PGS and MMD metrics. Unlike MMD, PGS is a bounded metric in $[0, 1]$. This means it can saturate, or reach its maximum value, when perturbations are too large and the distributions become non-overlapping. To account for this, we first determine the perturbation magnitude at which PGS saturates (specifically, exceeds $0.95$). We then apply perturbations only within this non-saturating range and compute the Spearman correlation between the metric scores and perturbation magnitudes. We visualize these correlation coefficients in Fig. 3, where each data point represents a combination of dataset, perturbation type, and metric. Our results show that PGS consistently exhibits a strong *rank* correlation with perturbation magnitude, comparable to that of MMD metrics. We note that while the degree-based and GIN-based MMD metrics struggle to detect the edge-swapping perturbation, PGS remains robust by leveraging multiple descriptors that compensate for compromised ones.

We provide more details in Appendix H, where we illustrate the behavior of PGS as a function of perturbation magnitude for all combinations of datasets and perturbations. From that analysis, we conclude that no single descriptor dominates the others across all combinations of datasets and perturbation types, underlining the necessity of considering a diverse set of graph descriptors. We present additional experiments for a PGS estimating the Total Variation distance in Appendix F.2. Appendix J provides similar results for a PGS variant using logistic regression instead of TabPFN.

### 5.3 PolyGraphScore Correlates With Model Quality

To demonstrate practical utility, we evaluate PGS on DiGress (Vignac et al., 2023), a state-of-the-art GGM, using denoising iterations and training epochs as proxies for model quality. PGS strongly correlates with both, capturing model improvement more faithfully than MMD metrics while maintaining a strong linear correlation with the percentage of valid graphs generated. All metrics were computed comparing 2048 reference graphs against 2048 generated graphs.

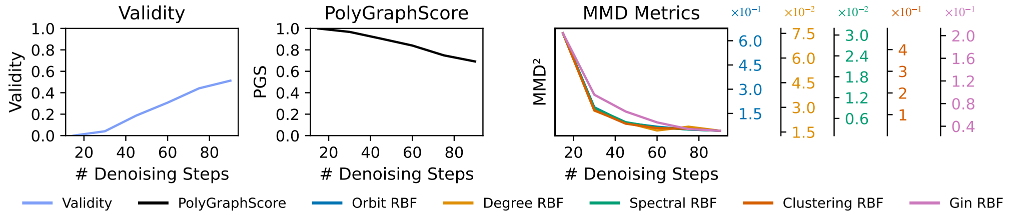


Figure 4: Trajectory of validity, PGS, and MMDs when increasing the number of denoising steps in DiGress on PLANAR-L.

Table 2: Negative Pearson correlation (↑) of validity with other distance-based metrics. Denoising refers to the experiments in which we vary the number of denoising iterations. Training refers to the experiments in which we monitor performance metrics during the training of DiGress models.

| | | PGS | Orbit RBF | Deg. RBF | Spec. RBF | Clust. RBF | GIN RBF |
|---|---|---|---|---|---|---|---|
| Denoising | PLANAR-L | **99.52** | 73.49 | 70.79 | 73.34 | 71.48 | 82.78 |
| Training | PLANAR-L | **99.05** | 84.33 | 76.52 | 79.05 | 81.61 | 81.07 |
| | SBM-L | **88.07** | 51.05 | 15.77 | 36.76 | 83.97 | 14.12 |
| | LOBSTER-L | **89.32** | -34.81 | -33.40 | -22.79 | 87.05 | -30.31 |

**Denoising Iterations.** We first analyze the impact of the number of denoising steps on sample quality. Six DiGress models are trained on the large procedural planar dataset using a range of 15 to 90 denoising steps. As shown in Fig. 4, increasing the number of steps generally improves model performance across all metrics. We find that PGS has a much stronger *linear* relationship with validity than MMD metrics, as shown by the *Pearson* correlation coefficients in Table 2. This tight relationship is especially encouraging as validity, alongside uniqueness and novelty, is often considered a gold standard metric for assessing model quality. Yet, validity is not always defined. Uniqueness and novelty can be provided jointly with PGS to offer complementary insights.

**Training Iterations.** Similarly, we assess the ability of MMD and PGS to track model quality throughout the training process on LOBSTER-L, PLANAR-L, and SBM-L. The central hypothesis is that reliable metrics should improve monotonically with training duration. We note that this relationship is non-linear, hence the use of Spearman's correlation coefficient. As illustrated for the SBM-L dataset in Fig. 5, PGS and validity align with this hypothesis, whereas MMD metrics exhibit erratic behavior. Analogous results for PLANAR-L and LOBSTER-L are provided in Appendix I. Spearman's rank correlation in Table 3 confirms this quantitatively across all datasets: both PGS and validity are strongly correlated with training duration, while MMD metrics show weak or even negative correlations. The Pearson correlations in Table 2 further show that *PGS maintains its strong linear correlation with validity* during training, a property not consistently shared by MMD metrics.

### 5.4 BENCHMARKING REPRESENTATIVE MODELS

We next present concrete PGS values and their associated subscores on a set of well-established models spanning distinct generative paradigms, including autoregressive architectures such as GRAN (Liao et al., 2019) and AutoGraph (Chen et al., 2025) and diffusion models such as ESGG (Bergmeister et al., 2023) and DIGRESS (Vignac et al., 2023). We benchmark them on our proposed datasets, SBM-L, LOBSTER-L, and PLANAR-L (with 2048 samples each, see Ap-

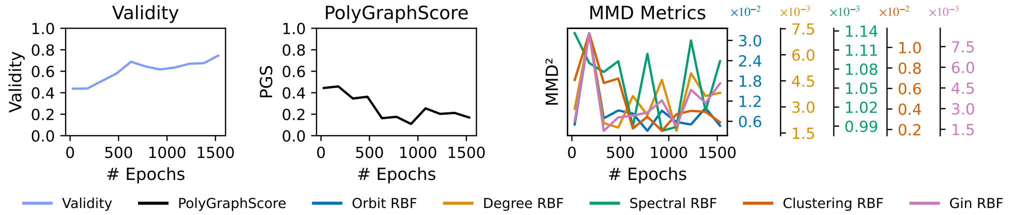


Figure 5: Trajectory of validity, PGS, and MMD metrics throughout training of DiGress on SBM-L.

Table 3: Sign-adjusted Spearman correlation (↑) of validity, PGS, and MMDs with the number of training iterations for DiGress.

| | Validity | PGS | Orbit RBF | Deg. RBF | Spec. RBF | Clust. RBF | GIN RBF |
|---|---|---|---|---|---|---|---|
| PLANAR-L | 92.31 | **93.71** | 86.71 | 41.96 | 83.22 | 67.83 | 81.82 |
| SBM-L | **83.64** | 62.73 | 20.00 | -19.09 | 18.18 | 58.18 | -38.18 |
| LOBSTER-L | **85.47** | 78.19 | -8.09 | -4.66 | 13.73 | 68.14 | -2.70 |

Table 4: Mean PGS ± standard deviation across synthetic and real-world graphs. AutoGraph* denotes a model pretrained on the PubChem dataset. More details can be found in the original paper (Chen et al., 2025). Values are multiplied by 100 for readability. Subscores are computed on the training set to select the best descriptor, and the final PGS refers to the score computed on the test set with the best descriptor.

| **Dataset** | **Model** | | | **PGS subscores** | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **VUN (↑)** | **PGS (↓)** | **Clust. (↓)** | **Deg. (↓)** | **GIN (↓)** | **Orb5. (↓)** | **Orb4. (↓)** | **Eig. (↓)** |
| PLANAR-L | AutoGraph | 85.1 | **34.0** ± 1.8 | **7.0** ± 2.9 | **7.8** ± 3.2 | **8.8** ± 3.0 | **34.0** ± 1.8 | **28.5** ± 1.5 | **26.9** ± 2.3 |
| | DIGRESS | 80.1 | 45.2 ± 1.8 | 24.8 ± 2.0 | 23.3 ± 1.2 | 29.0 ± 1.1 | 45.2 ± 1.8 | 40.3 ± 1.8 | 39.4 ± 2.0 |
| | GRAN | 1.6 | 99.7 ± 0.2 | 99.3 ± 0.2 | 98.3 ± 0.3 | 98.3 ± 0.3 | 99.7 ± 0.1 | 99.2 ± 0.2 | 98.5 ± 0.4 |
| | ESGG | **93.9** | 45.0 ± 1.4 | 10.9 ± 3.2 | 21.7 ± 3.0 | 32.9 ± 2.2 | 45.0 ± 1.4 | 42.8 ± 1.9 | 29.6 ± 1.6 |
| LOBSTER-L | AutoGraph | 83.1 | 18.0 ± 1.6 | 4.2 ± 1.9 | 12.1 ± 1.6 | 14.8 ± 1.5 | 18.0 ± 1.6 | 16.1 ± 1.6 | 13.0 ± 1.1 |
| | DIGRESS | **91.4** | **3.2** ± 2.6 | 2.0 ± 1.3 | **1.2** ± 1.5 | **2.3** ± 2.0 | **3.0** ± 3.1 | **4.5** ± 2.3 | **1.3** ± 1.1 |
| | GRAN | 41.3 | 85.4 ± 0.5 | 20.8 ± 1.1 | 77.1 ± 1.2 | 79.8 ± 0.6 | 85.4 ± 0.5 | 85.0 ± 0.6 | 69.8 ± 1.2 |
| | ESGG | 70.9 | 69.9 ± 0.6 | **0.0** ± 0.0 | 63.4 ± 1.1 | 66.8 ± 1.0 | 69.9 ± 0.6 | 66.0 ± 0.6 | 51.7 ± 1.8 |
| SBM-L | AutoGraph | **85.6** | **5.6** ± 1.5 | **0.3** ± 0.6 | **6.2** ± 1.4 | **6.3** ± 1.3 | **3.2** ± 2.2 | **4.4** ± 2.0 | **2.5** ± 2.2 |
| | DIGRESS | 73.0 | 17.4 ± 2.3 | 5.7 ± 2.8 | 8.2 ± 3.3 | 13.8 ± 1.7 | 17.4 ± 2.3 | 14.8 ± 2.5 | 8.7 ± 3.0 |
| | GRAN | 21.4 | 69.1 ± 1.4 | 50.2 ± 1.9 | 58.6 ± 1.4 | 69.1 ± 1.4 | 65.7 ± 1.3 | 62.8 ± 1.3 | 55.9 ± 1.5 |
| | ESGG | 10.4 | 99.4 ± 0.2 | 97.9 ± 0.5 | 97.5 ± 0.6 | 98.3 ± 0.4 | 96.8 ± 0.4 | 89.2 ± 0.7 | 99.4 ± 0.2 |
| Proteins | AutoGraph | - | **67.7** ± 7.4 | 47.7 ± 5.7 | 31.5 ± 8.5 | 45.3 ± 5.1 | **67.7** ± 7.4 | **47.4** ± 7.0 | 53.2 ± 6.9 |
| | DIGRESS | - | 88.1 ± 3.1 | **36.1** ± 4.3 | **29.2** ± 5.0 | **23.2** ± 5.3 | 88.1 ± 3.1 | 60.8 ± 3.6 | **23.4** ± 11.8 |
| | GRAN | - | 89.7 ± 2.7 | 86.0 ± 2.0 | 70.6 ± 3.1 | 71.5 ± 3.0 | 90.4 ± 2.4 | 84.4 ± 3.3 | 76.7 ± 4.7 |
| | ESGG | - | 79.2 ± 4.3 | 58.2 ± 3.6 | 54.0 ± 3.6 | 57.4 ± 4.1 | 80.2 ± 3.1 | 72.5 ± 3.0 | 24.3 ± 11.0 |

| **Dataset** | **Model** | | | **PGS subscores** | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **Valid (↑)** | **PGS (↓)** | **Topo (↓)** | **Morgan (↓)** | **ChemNet (↓)** | **MolCLR (↓)** | **Lipinski (↓)** |
| GUACAMOL | AutoGraph | 91.6 | 22.9 ± 0.5 | 8.2 ± 0.7 | 15.7 ± 0.8 | 22.9 ± 0.5 | 16.6 ± 0.4 | 19.4 ± 0.7 |
| | AutoGraph* | **95.9** | **10.4** ± 1.2 | **4.3** ± 0.7 | **4.7** ± 1.4 | **4.6** ± 0.6 | **1.7** ± 1.0 | **10.4** ± 1.2 |
| | DIGRESS | 85.2 | 32.7 ± 0.5 | 19.6 ± 0.6 | 20.4 ± 0.5 | 32.5 ± 0.7 | 22.9 ± 0.6 | 32.8 ± 0.5 |
| MOSES | AutoGraph | **87.4** | **29.6** ± 0.4 | **22.4** ± 0.4 | **16.3** ± 1.3 | **25.8** ± 0.7 | **20.5** ± 0.5 | **29.6** ± 0.4 |
| | DIGRESS | 85.7 | 33.4 ± 0.5 | 26.8 ± 0.4 | 24.8 ± 0.8 | 29.1 ± 0.6 | 24.3 ± 0.7 | 33.4 ± 0.5 |

pendix M) as well as the Proteins dataset with 92 samples (Dobson & Doig, 2003). Additionally, we present PGS benchmarks of AutoGraph and DiGress on the molecular datasets GuacaMol (Brown et al., 2019) and MOSES (Polykovskiy et al., 2020) using 10,000 generated samples for benchmarking. For these datasets, we propose domain-specific descriptors which we describe in Appendix C.2. Appendix K contains further benchmarking methodological details.

As shown in Table 4, AutoGraph and DiGress achieve the best overall PGS scores across most datasets. PGS generally aligns with VUN or validity rankings, though some exceptions exist—ESGG ranks highest in VUN on PLANAR-L but performs worse in PGS. The Proteins dataset yields the highest scores, suggesting greater modeling difficulty. Max-reduction proves helpful in edge cases like LOBSTER-L, where clustering coefficients are uniformly zero, preventing a single uninformative subscore from masking other structural flaws. When interpreting the final PGS score, note it can differ from individual subscores since they use different datasets. Subscores are averaged over cross-validation splits on the *training set* to select the most informative descriptor, while the final PGS is computed on the *test set*, potentially yielding different results. Appendix K compares MMD and PGS values using Gaussian TV pseudo-kernels (Table 11) and optimized RBF kernels (Tables 12 and 13). Overall, PGS yields more interpretable model rankings than MMDs.

We also consider a feature concatenation variant of PGS as an alternative to max-reduction, where we concatenate all descriptors and apply PCA to fit TabPFN's feature limits (500 for v2.0) in Appendix Q. While this yields tighter bounds (higher JSD estimates), it prevents identifying the most informative descriptor; therefore, we recommend max-reduction in practice.

## 6 Conclusion

We introduce PGS, a classifier-based evaluation that yields unit-scale metrics by training a discriminator on standard graph descriptors and selecting the most informative one. Instantiated with TabPFN to estimate the JS distance, PGS is fast and tuning-free. Across perturbation and model-quality studies, PGS increases monotonically with synthetic noise and correlates strongly–and often linearly–with validity and training progress. It also produces robust rankings with descriptor-specific subscores. To standardize GGM evaluation and model selection, we release the `PolyGraph` library, PGS, and the larger datasets, which we show are necessary to avoid high bias and variance observed in evaluation metrics. We discuss potential limitations in Appendix A. We hope that our work catalyzes progress in graph generation and, more broadly, enables effective evaluations of generative models where multiple combinations of possibly complementary descriptors are required.

## Ethics Statement

This work focuses on the development of evaluation methods for graph generative models. Our study does not involve human subjects, animals, or personal data. We do not foresee harm to individuals, groups, or the environment.

## Reproducibility Statement

To ensure reproducibility, we will publicly release the `PolyGraph` library that implements the `PolyGraphScore`, MMD metrics, and datasets discussed in this paper. This library is also provided to the reviewers as anonymized supplementary material. Unit tests ensure consistency of the MMD metrics implemented in `PolyGraph` with the implementations of Liao et al. (2019); Thompson et al. (2022). We refer to Appendices B and C for a more detailed explanation of the PGS estimation procedure and the graph descriptors considered in this work. Appendix M details how to generate our improved procedural datasets. We are committed to storing all data generated in this work (including model checkpoints and computed metrics) in a long-term private archive with a minimum guaranteed access period of ten years.

## References

Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning (ICML)*, 2017.

Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

Andreas Bergmeister, Karolis Martinkus, Nathanaël Perraudin, and Roger Wattenhofer. Efficient and scalable graph generation through iterative local expansion. In *International Conference on Learning Representations (ICLR)*, 2023.

Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations*, 2018.

Aleksandar Bojchevski, Oleksandr Shchur, Daniel Zügner, and Stephan Günnemann. Netgan: Generating graphs via random walks. In *International Conference on Machine Learning (ICML)*, pp. 609–618, 2018.

Karsten M Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *Fifth IEEE international conference on data mining (ICDM 2005)*, pp. 8–pp. IEEE, 2005.

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

Dexiong Chen, Markus Krimmel, and Karsten M. Borgwardt. Flatten graphs as sequences: Transformers are scalable graph generators. *CoRR*, abs/2502.02216, 2025. doi: 10.48550/ARXIV. 2502.02216. URL `https://doi.org/10.48550/arXiv.2502.02216`.

Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *J. Mol. Biol.*, 330(4):771–783, July 2003.

Dominik Maria Endres and JE Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 2003.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Kristen Grauman and Trevor Darrell. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research (JMLR)*, 8(4), 2007.

Arthur Gretton and Wittawat Jitkrittum. Openreview comment on ”revisiting classifier two-sample tests”. https://openreview.net/forum?id=SJkXfE5xx&noteId=ry-le414x, 2016. Accessed: 2025-07-22.

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

Aric Hagberg, Pieter J Swart, and Daniel A Schult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30. Curran Associates, Inc., 2017.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 01 2025. doi: 10.1038/s41586-024-08328-6.

Tomáš Hočevar. ORCA: ORbit Counting Algorithm. https://github.com/thocevar/orca, 2025. GitHub repository, GPL-3.0 License, last accessed: 2025-09-16.

Daniel Jiwoong Im, He Ma, Graham W. Taylor, and Kristin Branson. Quantitatively evaluating gans with divergences proposed for training. In *International Conference on Learning Representations (ICLR)*, 2018.

Marco Jiralerspong, Avishek Joey Bose, Ian Gemp, Chongli Qin, Yoram Bachrach, and Gauthier Gidel. Feature likelihood score: Evaluating generalization of generative models using samples, 2023.

Filip Ekström Kelvinius, Oskar B. Andersson, Abhijith S Parackal, Dong Qian, Rickard Armiento, and Fredrik Lindsten. Wyckoffdiff – a generative diffusion model for crystal symmetry. In *International Conference on Machine Learning (ICML)*, 2025.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A metric for evaluating music enhancement algorithms. *Proc. Interspeech*, 2018.

Markus Krimmel, Jenna Wiens, Karsten M. Borgwardt, and Dexiong Chen. Towards fast graph generation via autoregressive noisy filtration modeling. *CoRR*, abs/2502.02415, 2025. doi: 10.48550/ARXIV.2502.02415. URL https://doi.org/10.48550/arXiv.2502.02415.

Yujia Li, Kevin Swersky, and Richard S. Zemel. Generative moment matching networks. In *International Conference on Machine Learning (ICML)*, 2015.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, William L. Hamilton, David Duvenaud, Raquel Urtasun, and Richard S. Zemel. Efficient graph generation with graph recurrent attention networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4257–4267, 2019.

Chia-Cheng Liu, Harris Chan, and Kevin Luk. Auto-regressive graph generation modeling with improved evaluation methods. Presented at the NeurIPS Workshop on Graph Representation Learning, 2019. URL `https://grlearning.github.io/papers/77.pdf`. Workshop paper, Accessed: 2025-07-26.

Gang Liu, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph diffusion transformers for multi-conditional molecular generation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 8065–8092. Curran Associates, Inc., 2024.

David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *International Conference on Learning Representations (ICLR)*, 2017.

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 (ICML)*, 2022.

Andreas Mayr, Günter Klambauer, Thomas Unterthiner, Marvin Steijaert, Jörg K Wegner, Hugo Ceulemans, Djork-Arné Clevert, and Sepp Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chem Sci*, 9(24):5441–5451, June 2018.

XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Trans. Inf. Theory*, 56(11):5847–5861, 2010. doi: 10.1109/TIT.2010.2068870. URL `https://doi.org/10.1109/TIT.2010.2068870`.

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

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*, 2020.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet chemnet distance: A metric for generative models for molecules in drug discovery. *Journal of Chemical Information and Modeling*, 58(9):1736–1741, Sep 2018. ISSN 1549-9596.

RDKit. Rdkit: Open-source cheminformatics. `https://www.rdkit.org`, 2024. [Online; accessed 15-September-2025. Version 2024.09.6].

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Mag.*, 29(3):93–106, 2008. doi: 10.1609/AIMAG.V29I3.2157. URL `https://doi.org/10.1609/aimag.v29i3.2157`.

Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research (JMLR)*, 12(9), 2011.

Hamed Shirzad, Kaveh Hassani, and Danica J. Sutherland. Evaluating graph generative models with contrastively learned features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Vignesh Ram Somnath, Charlotte Bunne, Connor Coley, Andreas Krause, and Regina Barzilay. Learning graph models for retrosynthesis prediction. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 9405–9415. Curran Associates, Inc., 2021.

Joshua Southern, Jeremy Wayland, Michael M. Bronstein, and Bastian Rieck. Curvature filtrations for graph generative model evaluation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Bharath K. Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Gert R. G. Lanckriet, and Bernhard Schölkopf. Kernel choice and classifiability for RKHS embeddings of probability distributions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2009.

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

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation. In *Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*, 2019.

Aad W. van der Vaart. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998. See Section 10.2 for the Bernstein–von Mises theorem.

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

Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.*, 4(3):279–287, 2022.

Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International Conference on Machine Learning (ICML)*, 2018.

# Appendix

## CONTENTS

# A LIMITATIONS

Here, we touch upon some of the limitations of this work.

**Descriptor dependence and information loss.** PGS operates on hand-crafted descriptors rather than raw graphs. It therefore yields a lower bound of the divergence between *descriptor distributions*, which itself is a lower bound of the divergence between the *graph distributions*. If the divergence between descriptor distributions does not tightly approximate the divergence between graph distributions, the PGS is also inherently a loose bound on the divergence between graph distributions. This highlights the importance of considering expressive descriptors.

The final max-reduction can also under-utilize complementary signals across descriptors. This could be addressed by combining features prior to TabPFN fitting, and using TabPFN extensions for automatic feature selection[3]. If future TabPFN-like foundation models also support more input features, this limitation will vanish.

**Sample-size dependence** On the one hand, PGS requires several hundred samples to get an accurate metric value, as indicated in Appendix L, which requires some computational burden, especially for difficult-to-compute descriptors. On the other hand, the sample size used in our formulation of PGS is constrained by TabPFN's recommended 10k training limit, though this restriction is only an implementation detail. This might be problematic in practice if a large number of samples is required to obtain a tight bound. We recommend that users assess the variance of PGS carefully when considering new descriptors, graph types, and discriminators. The TabPFN extensions package also implements some approaches to extend the training size via subsampling and ensembling[4].

**Limited feature dimensionality.** While MMD can operate on high-dimensional graph descriptors, the classifier used in PGS may impose limits on the dimensionality of these features. The TabPFN model that we use in our work has been shown to be effective on up to 500-dimensional features. The graph descriptors proposed in previous works (c.f. Appendix C.1) are well within these limits. In the context of evaluating molecule generative models, we employ random projections to map 512-dimensional graph representations to a more compact feature space (c.f. Appendix C.2). A more sophisticated feature selection process may yield tighter bounds on the JS distance. We leave the exploration of optimal feature selection to future work.

**Scopes of graph types, datasets, and models.** Our experiments focus on common procedural datasets (with specific parameters), proteins, and molecules. We do not evaluate directed, temporal, or heterogeneous graphs, and leave this to future work. While we benchmark four different GGMs, covering autoregressive and denoising diffusion paradigms, we hope that future works adopt the PGS framework to extend benchmarks to a wider variety of methods.

**Application to other domains.** We focus on applying PGS to generative graph evaluation, where the need for rigorous assessment is particularly acute. Nonetheless, the same approach could extend to other domains, though we leave this unexplored. One promising direction is improving InceptionV3-style scoring: our multi-descriptor strategy could mitigate the sensitivity of FID to network initialization by max-reducing across multiple InceptionV3 initializations, which was shown to be problematic by Barratt & Sharma (2018).

# B PGS PSEUDOCODE

We provide pseudocode for the computation of PGS in Algorithm 1. We note that the procedure `estimate_divergence` corresponds to the algorithm we describe in Section 4.1 while `polygraphscore` implements the combination of descriptors we outline in Section 4.2.

[3] `https://github.com/PriorLabs/tabpfn-extensions`

[4] see Footnote 3

**Algorithm 1** PGS computation

```
procedure estimate_divergence(train, val, mode)
    clf ← fit_tabpfn(train)
    preds ← clf.predict(val.x)
    if mode = "jsd" then
        metric ← $\sqrt{\max(\texttt{log\_likelihood}(\text{preds}, \text{val.y}), 0)}$
    else
        $\gamma$ ← max_info_threshold(clf.predict(train.x), train.y)
        metric ← informedness(preds, val.y, $\gamma$)
    return metric

procedure train_test_divergence(reference, generated, descriptor, mode, $k$)
    ref_train, ref_test ← reference[0 :: 2], reference[1 :: 2]            ▷ Split reference graphs
    gen_train, gen_test ← generated[0 :: 2], generated[1 :: 2]            ▷ Split generated graphs
    $(X,\ Y)$ ← (descriptor(ref_train ∥ gen_train), [0 . . . 0, 1 . . . 1])
    folds ← stratified_folds($X, Y, k$)
    cv_metric ← 0
    for train, val ∈ folds do
        cv_metric ← cv_metric + estimate_divergence(train, val, mode)
    cv_metric ← cv_metric/$k$
    $(X_{\text{test}},\ Y_{\text{test}})$ ← (descriptor(ref_test ∥ gen_test), [0 . . . 0, 1 . . . 1])
    test_metric ← estimate_divergence($(X,\ Y), (X_{\text{test}},\ Y_{\text{test}})$, mode)
    return cv_metric, test_metric

procedure polygraphscore(ref, gen, mode)
    all_descriptors ← [orbit4, orbit5, deg, clust, spec, gin]
    all_metrics ← hash_map()
    for $d$ ∈ all_descriptors do
        all_metrics[$d$] ← train_test_divergence(ref, gen, $d$, mode, $k = 4$)
    best_desc ← $\arg\max_d$ all_metrics[$d$].cv_metric
    return all_metrics[best_desc].test_metric
```

# C GRAPH DESCRIPTORS

In this section, we discuss the vectorial graph descriptions used in our work. In Appendix C.1, we provide details on the descriptors we apply to the synthetic datasets (PLANAR-L, SBM-L, LOBSTER-L) and the Proteins dataset. These descriptors are, for the most part, identical to established descriptors introduced for MMD evaluations (You et al., 2018; Liao et al., 2019; Thompson et al., 2022). In Appendix C.2, we introduce novel descriptors for evaluating generative models for molecules.

We recommend that practitioners use domain-specific and expressive descriptors whenever possible, similar to our procedure for molecules in Appendix C.2. As discussed previously, one should aim to maximize the PGS metric when engineering graph descriptors.

## C.1 GENERIC DESCRIPTORS

We use graph descriptors that have previously been proposed for evaluations via Maximum Mean Discrepancy. Histograms of clustering coefficients and node degrees, as well as 4-node orbit counts, have been proposed by You et al. (2018). These descriptors were extended by Liao et al. (2019) via the spectrum of the graph Laplacian. Finally, Thompson et al. (2022) proposed to featurize graphs via randomly initialized GIN models. We extend these descriptors with 5-node orbit counts, computed with the ORCA algorithm (Hočevar, 2025). In our model benchmarks, we find that 5-node orbit counts oftentimes yield the highest PGS, hence representing a strong descriptor (c.f. Table 4). However, we find in the perturbation experiments (c.f. Appendix H) that no single descriptor consistently dominates the others. This demonstrates the importance of considering a wide variety of graph featurizers. We summarize our descriptors in Table 5.

Table 5: Generic graph descriptors.

| Descriptor | Meaning | Reference |
|---|---|---|
| Clust. | Histogram of clustering coefficients, discretized to 100 bins in $[0, 1]$ | You et al. (2018) |
| Deg. | Histogram of node degrees | You et al. (2018) |
| GIN | Activations of a randomly initialized GIN graph neural network | Thompson et al. (2022) |
| Eig. | Histogram of Laplacian spectrum, discretized to 200 bins in $[-10^{-5}, 2]$ | Liao et al. (2019) |
| Orb. 4 | 4-node orbit counts | You et al. (2018); Hočevar (2025) |
| Orb. 5 | 5-node orbit counts | Hočevar (2025) |

## C.2 MOLECULE-SPECIFIC DESCRIPTORS

We propose several novel descriptors for evaluating generative models for molecules via the PolyGraphScore framework. Some of these descriptors are established in chemoinformatics and are computed via RDKit (RDKit, 2024). Namely, topological quantities (`rdkit.Chem.GraphDescriptors`), physico-chemical parameters (`rdkit.Chem.Lipinski`) and classical Morgan molecule fingerprints (`rdkit.Chem.AllChem.GetMorganGenerator`). Additionally, we use learned representations extracted either from a SMILES-based LSTM model (Mayr et al., 2018) (termed ChemNet), or from the contrastively trained MolCLR graph neural network (Wang et al., 2022). The SMILES-based model has previously been used to formulate the Fréchet ChemNet distance (Preuer et al., 2018). To obtain more compact features, we map the learned representations into a 128-dimensional space via sparse random projections with a fixed random seed.

These descriptors can only be computed for molecular graphs which can be converted into `rdkit.Chem.rdchem.Mol` objects, i.e., for graphs which are chemically valid. Hence, we must filter generated graphs before computing a PGS score. A similar approach has been taken in the Fréchet ChemNet distance.

We summarize these descriptors in more detail in Table 6.

Table 6: Descriptors used for molecular graphs.

| Descriptor | Meaning | Features | Reference |
|---|---|---|---|
| Morgan | 128-D Morgan count fingerprint | Substructure hash counts | RDKit (2024) |
| ChemNet | 128-D projection of ChemNet embedding of canonical SMILES string | Latent | Mayr et al. (2018) |
| MolCLR | 128-D projection of MolCLR embedding of molecule graph | Latent | Wang et al. (2022) |
| Topo | Topological/topochemical descriptors based on the bond structure | 1. `AvgIpc`<br>2. `BertzCT`<br>3. `BalabanJ`<br>4. `HallKierAlpha`<br>5. `Kappa1`<br>6. `Kappa2`<br>7. `Kappa3`<br>8. `Chi0`<br>9. `Chi0n`<br>10. `Chi0v`<br>11. `Chi1`<br>12. `Chi1n`<br>13. `Chi1v`<br>14. `Chi2n`<br>15. `Chi2v`<br>16. `Chi3n`<br>17. `Chi3v`<br>18. `Chi4n`<br>19. `Chi4v` | RDKit (2024) |
| Lipinski | Structural and physico-chemical parameters | 1. `HeavyAtomCount`<br>2. `NHOHCount`<br>3. `NOCount`<br>4. `NumHAcceptors`<br>5. `NumHDonors`<br>6. `NumHeteroatoms`<br>7. `NumRotatableBonds`<br>8. `RingCount`<br>9. `NumAliphaticCarbocycles`<br>10. `NumAliphaticHeterocycles`<br>11. `NumAliphaticRings`<br>12. `NumAromaticCarbocycles`<br>13. `NumAromaticHeterocycles`<br>14. `NumAromaticRings`<br>15. `NumHeterocycles`<br>16. `NumSaturatedCarbocycles`<br>17. `NumSaturatedHeterocycles`<br>18. `NumSaturatedRings`<br>19. `NumAmideBonds`<br>20. `NumAtomStereoCenters`<br>21. `NumUnspecifiedAtomStereoCenters`<br>22. `NumBridgeheadAtoms`<br>23. `NumSpiroAtoms`<br>24. `FractionCSP3`<br>25. `Phi` | RDKit (2024) |

## D MMD as Linear Classification Risk

In this section, we expand on the discussion in Section 3.1 and derive how MMD may be seen as the optimal risk for distinguishing between $P$ and $Q$ of a binary classifier in the reproducing kernel Hilbert space $\mathcal{H}$.

Using the notation $\mathbb{E}_x[k(x,\,\cdot\,)]$ for the Riesz representative of the (under mild conditions) bounded linear form $f \mapsto \mathbb{E}_x[\langle f, k(x,\,\cdot\,)\rangle]$, one may show:

$$\begin{aligned}\mathrm{MMD}(P,Q,k) &= \|\mathbb{E}_{x\sim P}[k(x,\,\cdot\,)] - \mathbb{E}_{y\sim Q}[k(y,\,\cdot\,)]\|_{\mathcal{H}} \\ &= \sup_{\|D\|_{\mathcal{H}}\leq 1} \langle D\,,\,\mathbb{E}_{x\sim P}[k(x,\,\cdot\,)] - \mathbb{E}_{y\sim Q}[k(y,\,\cdot\,)]\rangle \\ &= \sup_{\|D\|_{\mathcal{H}}\leq 1} \langle D\,,\,\mathbb{E}_{x\sim P}[k(x,\,\cdot\,)]\rangle - \langle D\,,\,\mathbb{E}_{y\sim Q}[k(y,\,\cdot\,)]\rangle \\ &= \sup_{\|D\|_{\mathcal{H}}\leq 1} \mathbb{E}_{x\sim P}[D(x)] - \mathbb{E}_{y\sim Q}[D(y)].\end{aligned} \tag{3}$$

We use the Cauchy-Schwarz inequality in the second equality, the linearity of the inner product in the third equality, and the definition of the Riesz representative in the last equality.

This framing reveals that MMD is precisely the optimal linear classification risk achievable by a discriminator $D$ in the unit ball of the function space induced by the kernel.

## E Background on $f$-Divergences and Total Variation Distance

Let $P$ and $Q$ be probability measures on $\mathcal{X}$ that are assumed to be absolutely continuous with respect to a base measure $\mu$, having densities $p$ and $q$. For now, also assume $P$ to be absolutely continuous w.r.t. $Q$. For a convex, lower-semicontinuous function $f : \mathbb{R}_+ \to \mathbb{R}$ satisfying $f(0) = 1$, the $f$-divergence of $P$ from $Q$ is defined as:

$$D_f(P \parallel Q) := \int_{\mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right) d\mu \tag{4}$$

As shown by Nguyen et al. (2010), $f$-divergence can be estimated via a variational objective similar to that of MMD. Using the Fenchel conjugate $f^*(v) := \sup_{u\in\mathbb{R}_+} uv - f(u)$, the $f$-divergence is lower-bounded by:

$$D_f(P \parallel Q) \geq \sup_{D\in\mathcal{F}} \mathbb{E}_{x\sim P}[D(x)] - \mathbb{E}_{y\sim Q}[f^*(D(x))], \tag{5}$$

for any family $\mathcal{F}$ of measurable functions $D : \mathcal{X} \to \mathbb{R}$. The bound is tight if and only if the functional class $\mathcal{F}$ is sufficiently expressive to contain a subderivative of $f$ at the density ratio $p(x)/q(x)$. Such a function then achieves the supremum. The variational formulation of the Jensen-Shannon divergence in Eqn. (2) is a special case of Eqn. (5)

**Total Variation Distance.** The total variation (TV) distance corresponds to $f(x) = \frac{1}{2}|1-x|$. One may easily verify that the integral in Eqn. (4) evaluates to half of the $L^1$ distance between $p$ and $q$. As we show in Appendix F.1, its variational objective in Eqn. (5) can be reduced to:

$$\sup_{\substack{D:\mathcal{X}\to[0,1]\\ \gamma\in[0,1]}} \mathbb{E}_{x\sim P}\left[[D(x) > \gamma]\right] - \mathbb{E}_{x\sim Q}[[D(x) > \gamma]], \tag{6}$$

where we use the Iverson bracket $[D(x) > \gamma]$ to denote the binarization of $D$ at the threshold $\gamma$. This objective is also known as the Informedness (or Youden's J statistic) of the discriminator $D$. It has a clear geometric interpretation as the maximum vertical distance between the ROC curve of $D$ and the chance diagonal, with a fixed scale of $[0, 1]$.

## F PGS-TV: Estimating Total Variation Distances

In this section, we propose an alternative variant of the PGS, using variational estimates of the total variation (TV) distance in place of the Jensen-Shannon distance. We term this variant PGS-TV.

We recall from Appendix E that the variational objective for the TV distance is given by the informedness of a dichotomized classifier. We provide a proof of this fact in Appendix F.1. When computing PGS-TV, the choice of binarization threshold is considered part of the fitting process of the classifier. Hence, we choose $\gamma$ to maximize the vertical distance of the ROC on the *fit* set. We refer to Appendix B for pseudocode. In Appendices F.2 and F.3, we present an empirical investigation of PGS-TV, analogous to the experiments presented in Sections 5.2 and 5.3. Finally, we discuss the advantages of PGS over PGS-TV in Appendix F.4

### F.1 Variational Formulation of TV Distance

One may easily verify that for $f(u) = \frac{1}{2}|1-u|$, we have the following Fenchel conjugate:

$$f^*(v) = \sup_{u \in \mathbb{R}_+} uv - \frac{1}{2}|1-u| = \begin{cases} -\frac{1}{2} & \text{if} \quad v < -\frac{1}{2} \\ v & \text{if} \quad v \in [-\frac{1}{2}, \frac{1}{2}] \\ \infty & \text{if} \quad v > \frac{1}{2} \end{cases} \tag{7}$$

We recall the variational lower bound:

$$D_{TV}(P \parallel Q) \geq \sup_{D \in \mathcal{F}} \mathbb{E}_{x \sim P}[D(x)] - \mathbb{E}_{y \sim Q}[f^*(D(x))] \tag{8}$$

Without weakening the lower bound, we may restrict ourselves to families of functions which are upper-bounded by $\frac{1}{2}$ almost everywhere w.r.t. $Q$. Indeed, discriminators $D$ that do not satisfy this have a variational bound of $-\infty$. Since we are assuming $P \ll Q$, the discriminators are then also upper-bounded almost everywhere w.r.t. $P$. Hence, w.l.o.g., we may assume that they are upper-bounded by $\frac{1}{2}$ everywhere. Under these assumptions, we obtain the simpler formulation:

$$\begin{aligned} D_{TV}(P \parallel Q) &\geq \sup_{D \in \mathcal{F}} \mathbb{E}_{x \sim P}[D(x)] - \mathbb{E}_{y \sim Q}\left[\max\left(D(x), -\frac{1}{2}\right)\right] \\ &= \sup_{D \in \mathcal{F}} \int_{\mathcal{X}} D(x)p(x) - \max\left(D(x), -\frac{1}{2}\right) q(x) d\mu \end{aligned} \tag{9}$$

Under the constraint that $D(x) \leq \frac{1}{2}$, we may maximize the expression above in a pointwise fashion by:

$$D(x) = \begin{cases} \frac{1}{2} & \text{if} \quad p(x) > q(x) \\ -\frac{1}{2} & \text{if} \quad p(x) \leq q(x) \end{cases} \tag{10}$$

We note that this is consistent with the finding of Nguyen et al. (2010) that $D(x)$ should attain a subderivative of $f$ at the point $\frac{p(x)}{q(x)}$. Therefore, without weakening the lower bound, we may write:

$$\begin{aligned} D_{TV}(P \parallel Q) &\geq \sup_{D: \mathcal{X} \to \left\{-\frac{1}{2}, \frac{1}{2}\right\}} \mathbb{E}_{x \sim P}[D(x)] - \mathbb{E}_{x \sim Q}\left[\max\left(D(x), -\frac{1}{2}\right)\right] \\ &= \sup_{D: \mathcal{X} \to \left\{-\frac{1}{2}, \frac{1}{2}\right\}} \mathbb{E}_{x \sim P}[D(x)] - \mathbb{E}_{x \sim Q}[D(x)] \\ &= \sup_{D: \mathcal{X} \to \{0,1\}} \mathbb{E}_{x \sim P}[D(x)] - \mathbb{E}_{x \sim Q}[D(x)] \\ &= \sup_{\substack{D: \mathcal{X} \to [0,1] \\ \gamma \in [0,1]}} \mathbb{E}_{x \sim P}[[D(x) > \gamma]] - \mathbb{E}_{x \sim Q}[[D(x) > \gamma]] \end{aligned} \tag{11}$$

The first equality is derived from the observation that $D(x) \geq -\frac{1}{2}$ always holds and the maximum is therefore redundant. The second equality is obtained by noting that the expression is invariant under the addition of constants to $D$ (in this case, we add $\frac{1}{2}$).

Without relying on the results of Nguyen et al. (2010), we now show that this bound is tight, even when $P \not\ll Q$. To work in this more general setting, we redefine the total variation distance as half the $L^1$ distance of $p$ and $q$:

$$D_{TV}(P \parallel Q) := \frac{1}{2}\|p - q\|_{L^1(\mathcal{X}, \mu)} = \frac{1}{2}\int_{\mathcal{X}} |p(x) - q(x)| d\mu \tag{12}$$

One may verify that this matches our original definition when $P \ll Q$. For any measurable set $A \subset \mathcal{X}$, we note that:

$$1 = \int_A p(x)d\mu + \int_{A^C} p(x)d\mu = \int_A q(x)d\mu + \int_{A^C} q(x)d\mu \tag{13}$$

Hence, rearranging, we obtain:

$$\int_A p(x) - q(x)d\mu = \int_{A^C} q(x) - p(x)d\mu \tag{14}$$

Defining $A := \{x \in \mathcal{X} \ : \ p(x) \geq q(x)\}$ and applying this identity, we get:

$$\begin{aligned} \frac{1}{2}\int_{\mathcal{X}} |p(x) - q(x)|d\mu &= \frac{1}{2}\int_A p(x) - q(x)d\mu + \frac{1}{2}\int_{A^C} q(x) - p(x)d\mu \\ &= \int_A p(x) - q(x)d\mu \end{aligned} \tag{15}$$

Since $A$ is exactly the set on which $p(x) - q(x)$ is non-negative, it is also clear that for any other $B \subset \mathcal{X}$, we have:

$$\frac{1}{2}\int_{\mathcal{X}} |p(x) - q(x)|d\mu \geq \int_B p(x) - q(x)d\mu \tag{16}$$

Thus, we may write:

$$\begin{aligned} D_{TV}(P \,\|\, Q) &= \sup_{B \subset \mathcal{X}} \int_B p(x) - q(x)d\mu \\ &= \sup_{D:\mathcal{X}\to\{0,1\}} \int_{\mathcal{X}} D(x)(p(x) - q(x))d\mu \\ &= \sup_{D:\mathcal{X}\to\{0,1\}} \mathbb{E}_{x\sim P}[D(x)] - \mathbb{E}_{x\sim Q}[D(x)] \end{aligned} \tag{17}$$

This is exactly the variational lower bound which we have derived above. Hence, we have shown it to be tight, even in the setting where $P \not\ll Q$.

### F.2 PGS-TV Tracks Synthetic Data Perturbations

We now present perturbation experiments for the PGS-TV variant that are analogous to those shown in Section 5.2.

We plot a summary of the Spearman correlation of the metrics with perturbation magnitude in Fig. 6. Compared to Fig. 3, we find that PGS-TV exhibits slightly lower correlations. Figs. 7 and 8 show the response of PGS-TV to perturbation over the entire and cropped magnitude range, respectively. For a more detailed explanation of this type of plot, we refer to Appendix H. From the plots we conclude that PGS-TV qualitatively exhibits the expected behavior of increasing with perturbation magnitude and eventually saturating. However, in some cases (e.g., edge addition on proteins), the PGS-TV flattens out, leading to lower correlations.

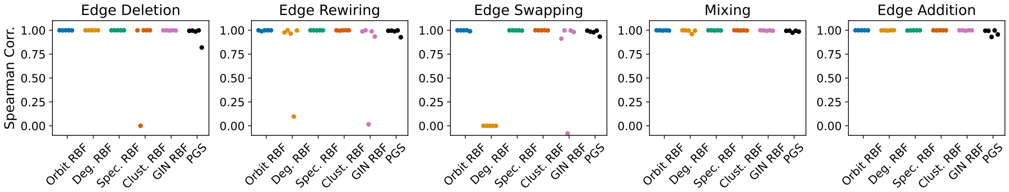


Figure 6: Spearman correlation of MMD metrics and PGS-TV with the magnitude of perturbation of datasets.

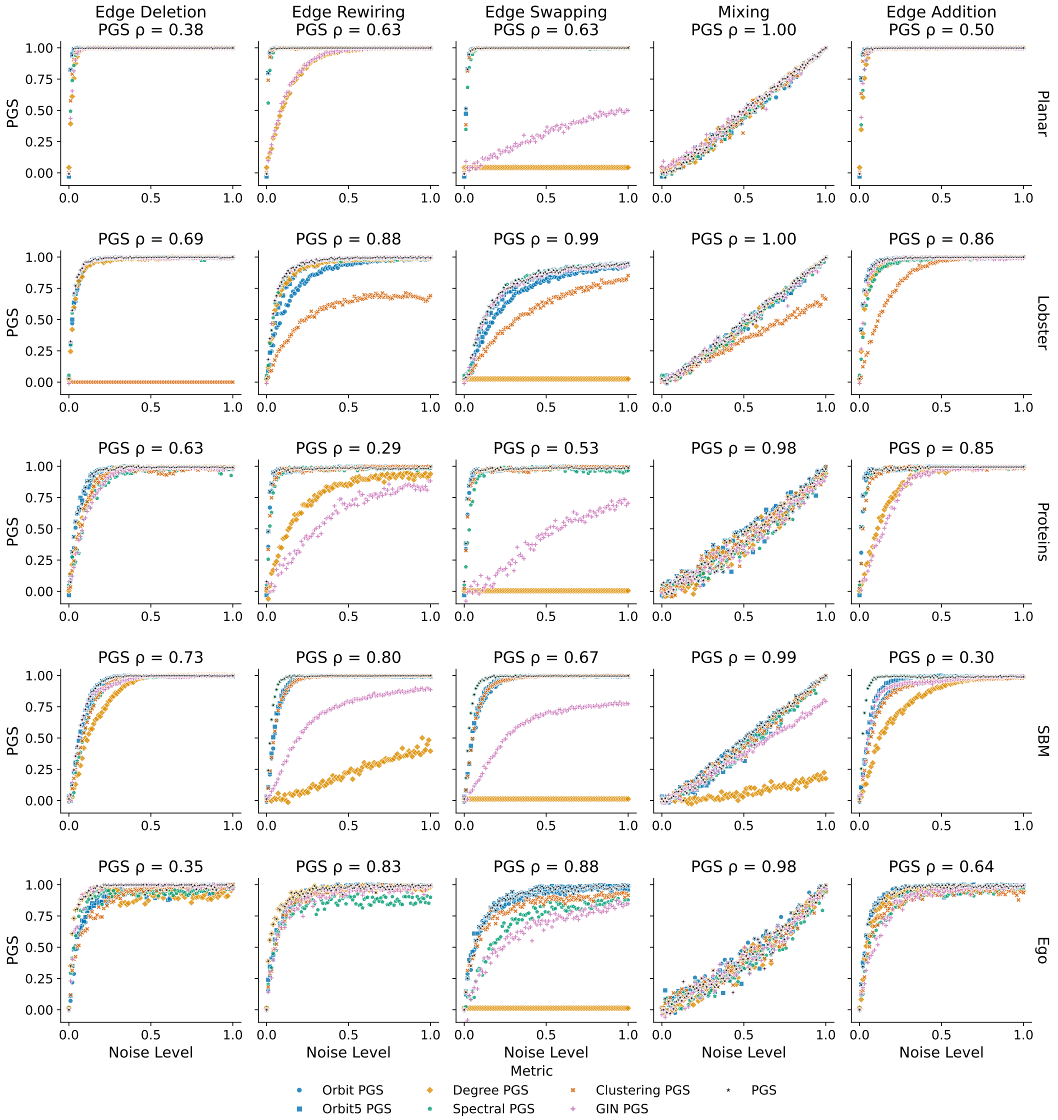


Figure 7: Behavior of descriptor-specific and aggregated PGS-TV as data distributions are perturbed. The perturbation type varies across rows while dataset varies across columns. The Spearman correlation of the aggregate PGS and the perturbation level is denoted by $\rho$.

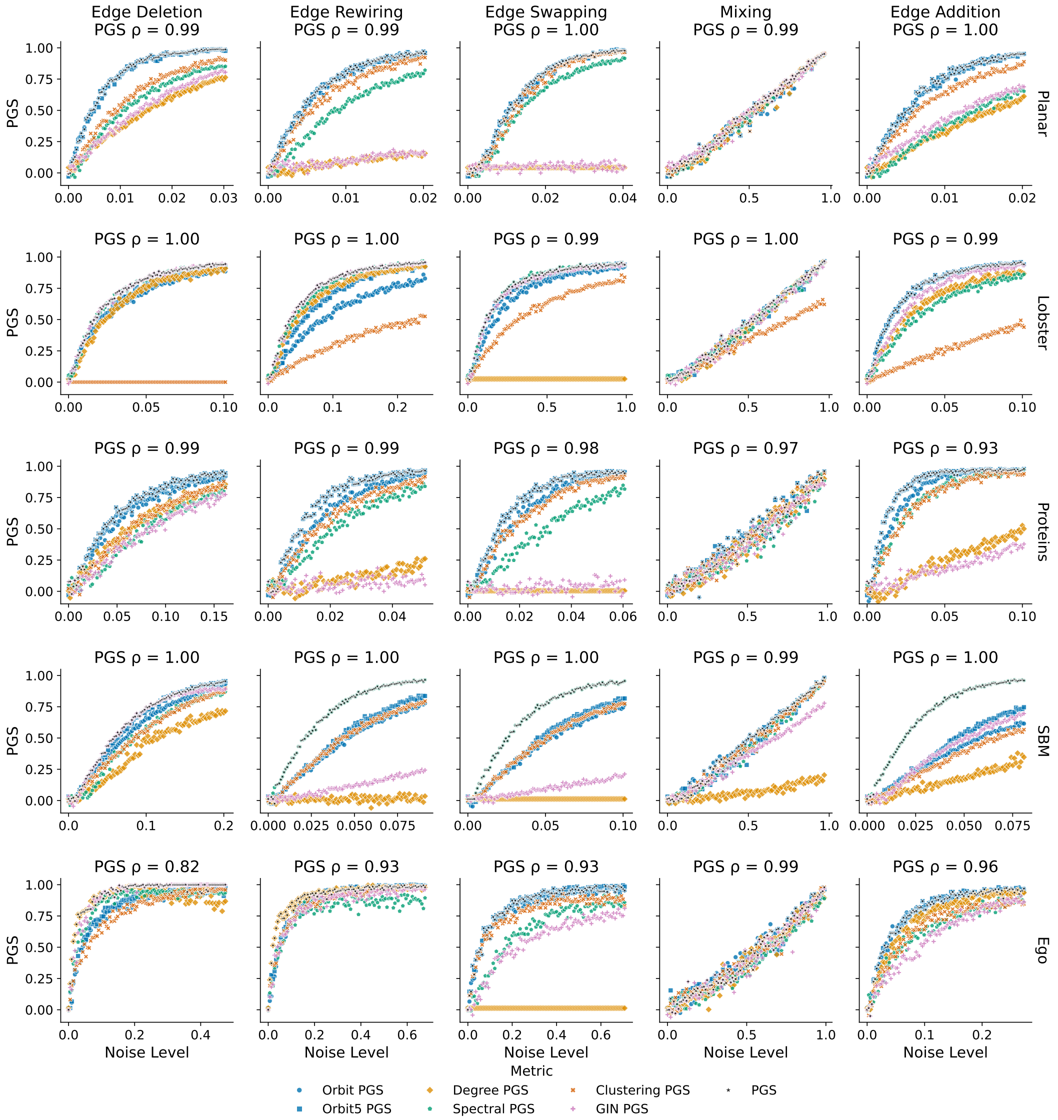


Figure 8: Behavior of descriptor-specific and aggregated PGS-TV as data distributions are perturbed. The perturbation type varies across rows, while the dataset varies across columns. The Spearman correlation of the aggregate PGS and the perturbation level is denoted by $\rho$.

### F.3 PGS-TV Correlates With Model Quality

Analogous to Section 5.3, we now investigate how the PGS-TV variant correlates with proxy variables of model quality. In Fig. 9, we illustrate how PGS-TV behaves as the number of denoising steps in DiGress is varied. As in Fig. 4, we find that PGS-TV correlates with validity in a highly linear fashion.

As in Section 5.3, we compute Pearson correlation coefficients between PGS-TV and validity. When varying the number of denoising steps, we find that PGS-TV exhibits a more linear relationship with validity than any of the MMD metrics.

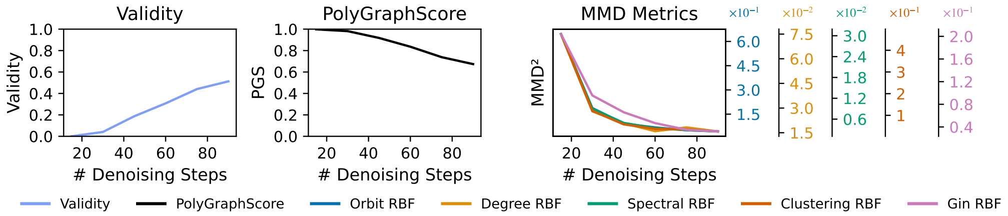


Figure 9: Behavior of validity, PGS-TV, and MMDs as the number of denoising steps in DiGress is varied on PLANAR-L.

We examine the behavior of PGS-TV throughout training in Fig. 9. Qualitatively, a clear positive relationship emerges between training duration and PGS-TV. This trend is confirmed quantitatively in Table 8, where Spearman correlation coefficients show that most MMD metrics often exhibit weak or negative correlations, while PGS-TV consistently correlates positively with training duration. However, this correlation is weaker than that of PGS-JS (see Table 3) and the clustering-based MMD metric. A similar pattern appears in Table 7 (bottom three rows): PGS-TV correlates reliably with validity, whereas most MMD metrics show inconsistent behavior. Nevertheless, in two out of three cases, the clustering-based MMD metric achieves a stronger correlation with validity than PGS-TV.

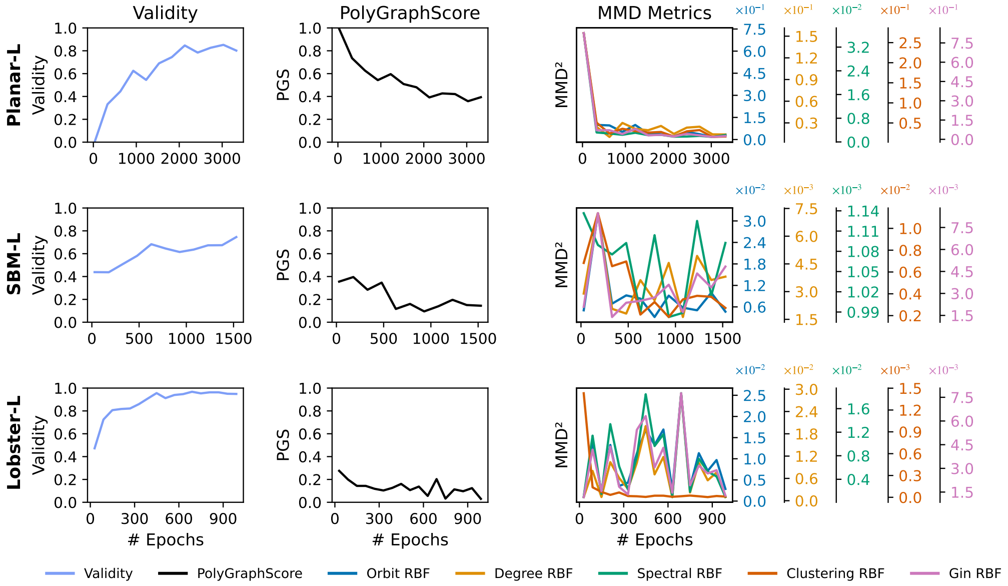


Figure 10: Behavior of validity, PGS-TV, and MMD metrics throughout training of DiGress on procedural graph datasets.

Table 7: Negative Pearson correlation (↑) of validity with other performance metrics. Denoising refers to the experiments in which we vary the number of denoising iterations. Training refers to the experiments in which we monitor performance metrics during the training of DiGress models.

| | | TV-PGS | Orbit RBF | Deg. RBF | Spec. RBF | Clust. RBF | GIN RBF |
|---|---|---|---|---|---|---|---|
| Denoising | Planar-L | **99.24** | 73.49 | 70.79 | 73.34 | 71.48 | 82.78 |
| Training | Planar-L | **99.42** | 84.33 | 76.52 | 79.05 | 81.61 | 81.07 |
| | SBM-L | **84.07** | 52.07 | 16.60 | 35.32 | 83.82 | 14.64 |
| | Lobster-L | 69.18 | -34.81 | -33.40 | -22.79 | **87.05** | -30.31 |

Table 8: Sign-adjusted Spearman correlation (↑) of validity, PGS-TV, and MMDs with number of training iterations of DiGress.

| | Validity | PGS | Orbit RBF | Deg. RBF | Spec. RBF | Clust. RBF | GIN RBF |
|---|---|---|---|---|---|---|---|
| Planar-L | 92.31 | **93.71** | 86.71 | 41.96 | 83.22 | 67.83 | 81.82 |
| SBM-L | **82.73** | 63.64 | 20.00 | -19.09 | 18.18 | 58.18 | -38.18 |
| Lobster-L | **85.47** | 62.91 | -8.09 | -4.66 | 13.73 | 68.14 | -2.70 |

### F.4 Comparison of PGS and PGS-TV

Overall, the experiments in Appendices F.2 and F.3 have demonstrated that PGS-TV is a viable alternative to the PGS metric we presented in the main paper, correlating to a high degree with synthetic data perturbations and proxy variables of model quality. Nevertheless, we found that PGS exhibits stronger correlations and appears like a more robust choice.

While we have no definite explanation for these observations, we hypothesize that the choice of binarization threshold in PGS-TV may introduce some noise into the estimate. Additionally, maximum likelihood classifiers (like logistic regression) inherently maximize the log-likelihood objective of the JS divergence. Bayesian inference (approximated by TabPFN) may be expected to behave similarly in the large sample size limit (van der Vaart, 1998). However, neither maximum likelihood estimation nor Bayesian inference directly optimizes the variational objective of the TV distance, i.e., informedness. This can lead to a misalignment when estimating the PGS-TV, potentially resulting in looser variational bounds.

For these reasons, we recommend using the PGS variant presented in the main paper, estimating lower bounds on the Jensen-Shannon distance.

## G Supplemental for: High Bias and Variance Plague MMD-based GGM Benchmarks

Here, we show that the conclusions of Section 5.1 expand to all combinations of models, descriptors, and datasets, and provide additional experimental details. All MMD estimates provided here and in Figs. 2a and 2b are RBF MMDs, as proposed by Thompson et al. (2022). The kernel is selected by taking the maximum over the bandwidths $\{\sigma_i\}_{i=1}^{1}0 = \{0.01, 0.1, 0.25, 0.5, 0.75, 1.0, 2.5, 5.0, 7.5, 10.0\}$.

Specifically, we subsampled 8 to 4096 graphs 100 times with replacement from a total of 8192 samples for the reference and generated graphs. We subsequently computed the median, 5$^{\text{th}}$ and 95$^{\text{th}}$ quantiles to estimate the variation of MMD. We computed such experiments for all model-generated samples we considered (ESGG, AutoGraph, DiGress and GRAN) and considered all descriptors (degree histogram, clustering histogram, orbit count for graphlet sizes 4 and 5, and the graph Laplacian eigenvalues) and all procedural datasets (SBM, Lobster and Planar).

Based on those findings, we introduce Planar-L, SBM-L, and Lobster-L, larger versions of the previously used datasets. Details for these new datasets are presented in Appendix M

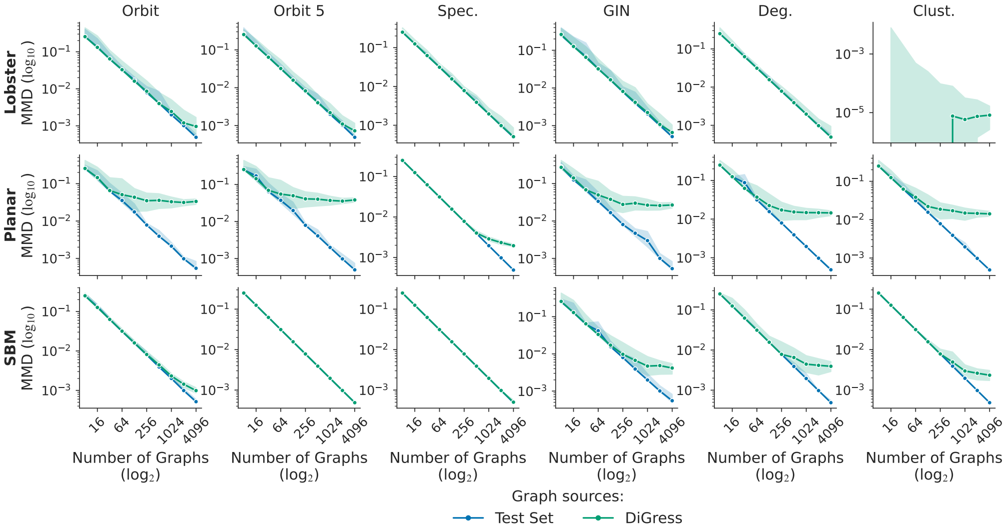


Figure 11: Behavior of biased MMD estimates as the number of samples is varied for DiGress.

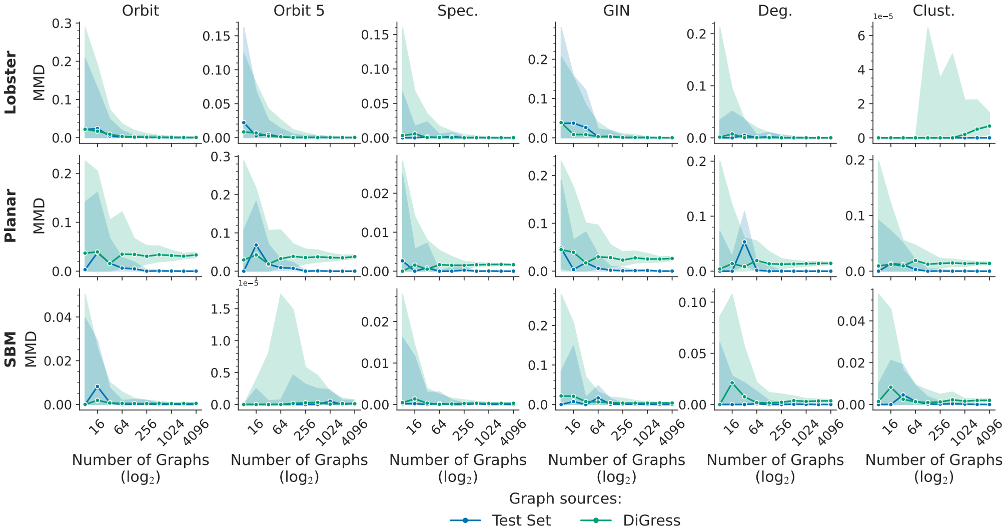


Figure 12: Behavior of unbiased MMD estimates as the number of samples is varied for DiGress.

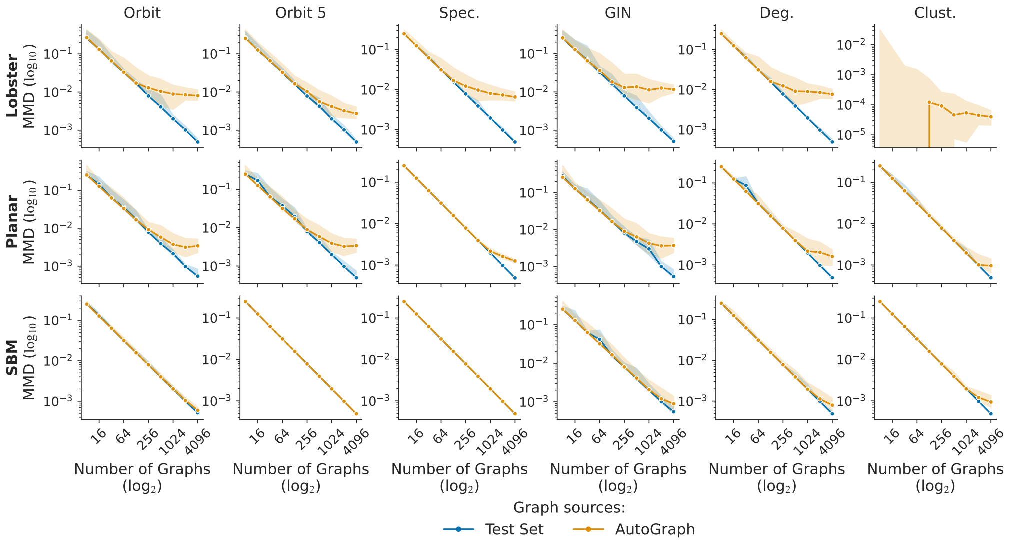


Figure 13: Behavior of biased MMD estimates as the number of samples is varied for AutoGraph.

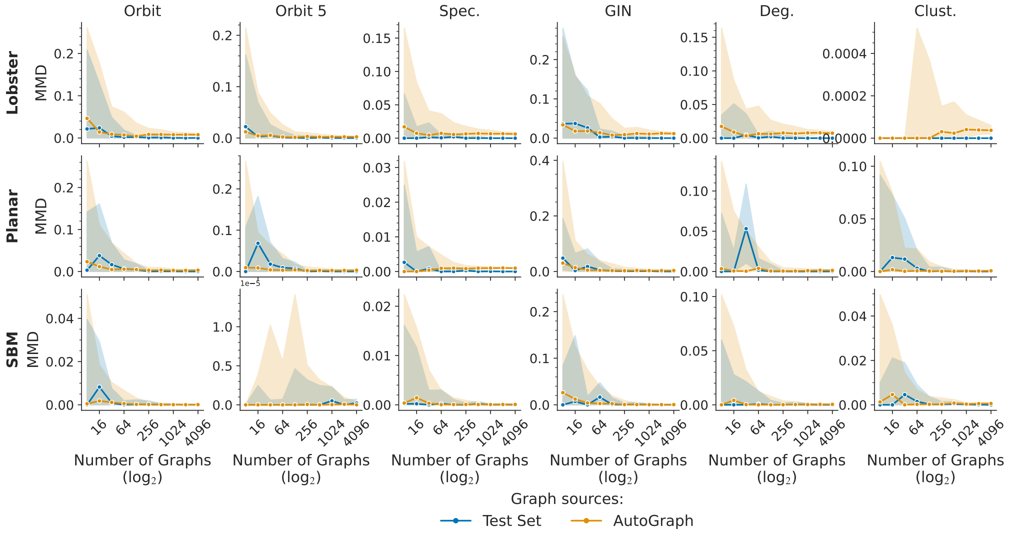


Figure 14: Behavior of unbiased MMD estimates as the number of samples is varied for AutoGraph.

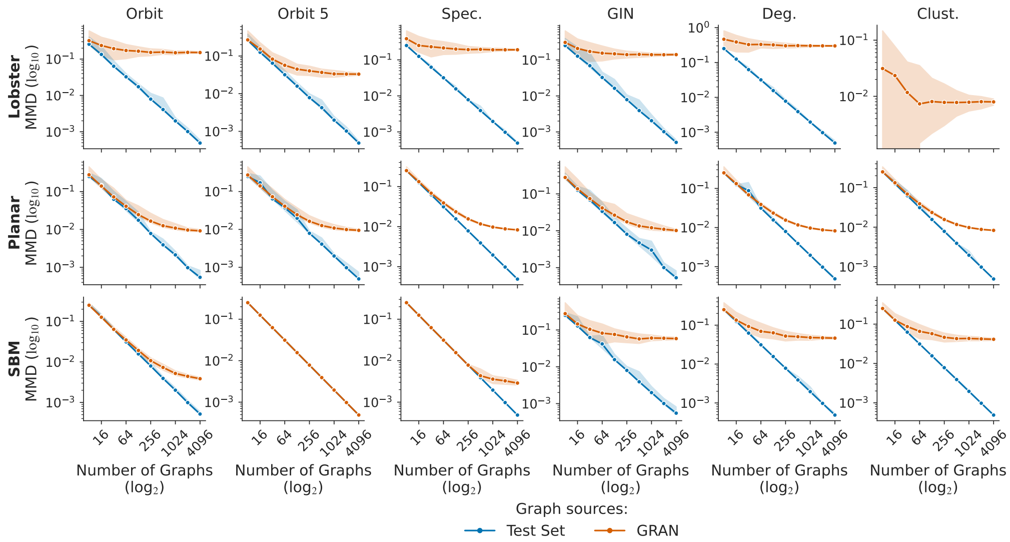


Figure 15: Behavior of biased MMD estimates as the number of samples is varied for GRAN.

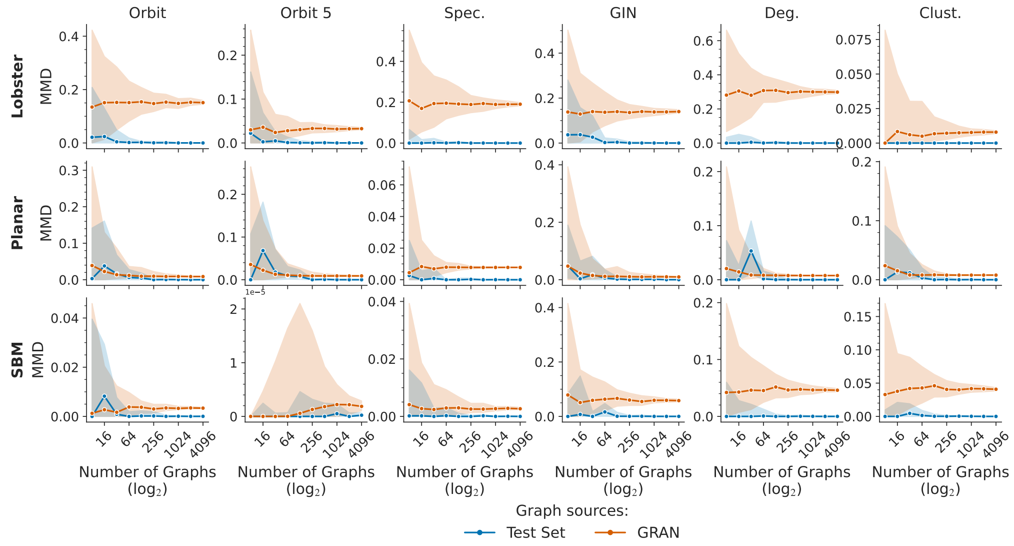


Figure 16: Behavior of unbiased MMD estimates as the number of samples is varied for GRAN.

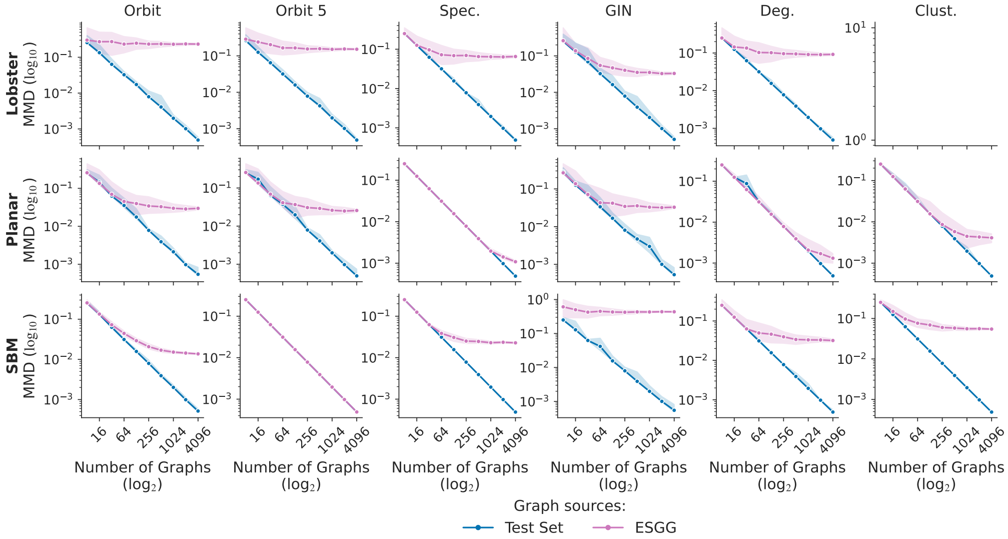


Figure 17: Behavior of biased MMD estimates as the number of samples is varied for ESGG.

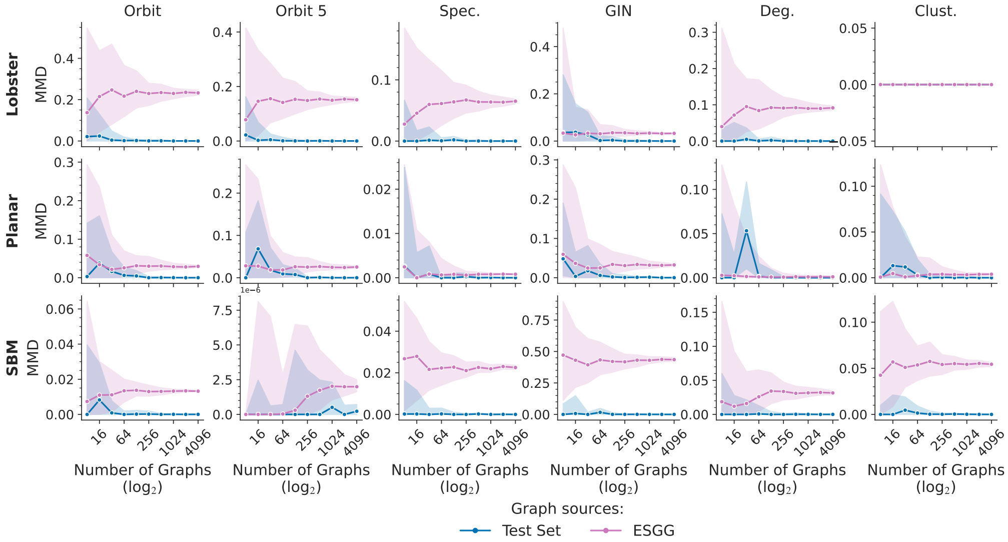


Figure 18: Behavior of unbiased MMD estimates as the number of samples is varied for ESGG.

# H Supplemental for: PGS Tracks Synthetic Data Perturbations

In this section, we provide further details for the experiments presented in Section 5.2. In particular, we illustrate in more detail how PGS responds to perturbations and present results for the TV variant.

In Fig. 19, we illustrate how PGS (descriptor-specific scores and the summary PGS) responds to various perturbations on different datasets. In this figure, we illustrate the response over the whole range of magnitudes $[0, 1]$. As anticipated, the PGS saturates quickly as the support of the perturbed distribution becomes disjoint from the support of the true data distribution. We note that the PGS consistently responds in a monotonic fashion to the magnitude of perturbation.

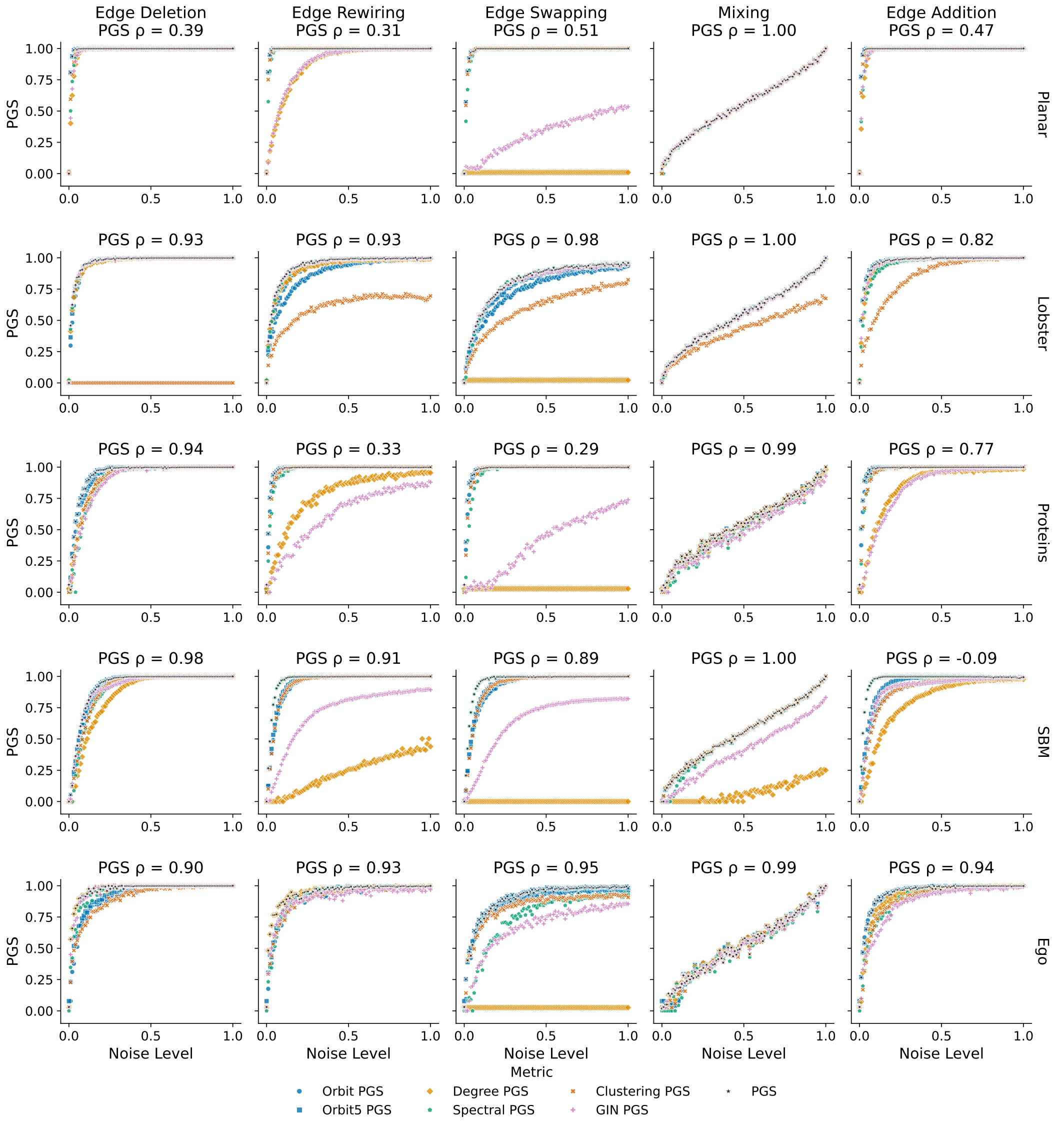


Figure 19: Behavior of descriptor-specific and aggregated PGS (JS) as data distributions are perturbed.

Based on the data from Fig. 19, we select a threshold for each combination of perturbation type and dataset at which the summary PGS saturates above 0.95. We illustrate the behavior of PGS-JS on these cropped ranges in Fig. 20.

We find that there is no single descriptor that consistently provides the tightest PGS estimate. This highlights the importance of evaluating many different descriptors when computing a PGS.

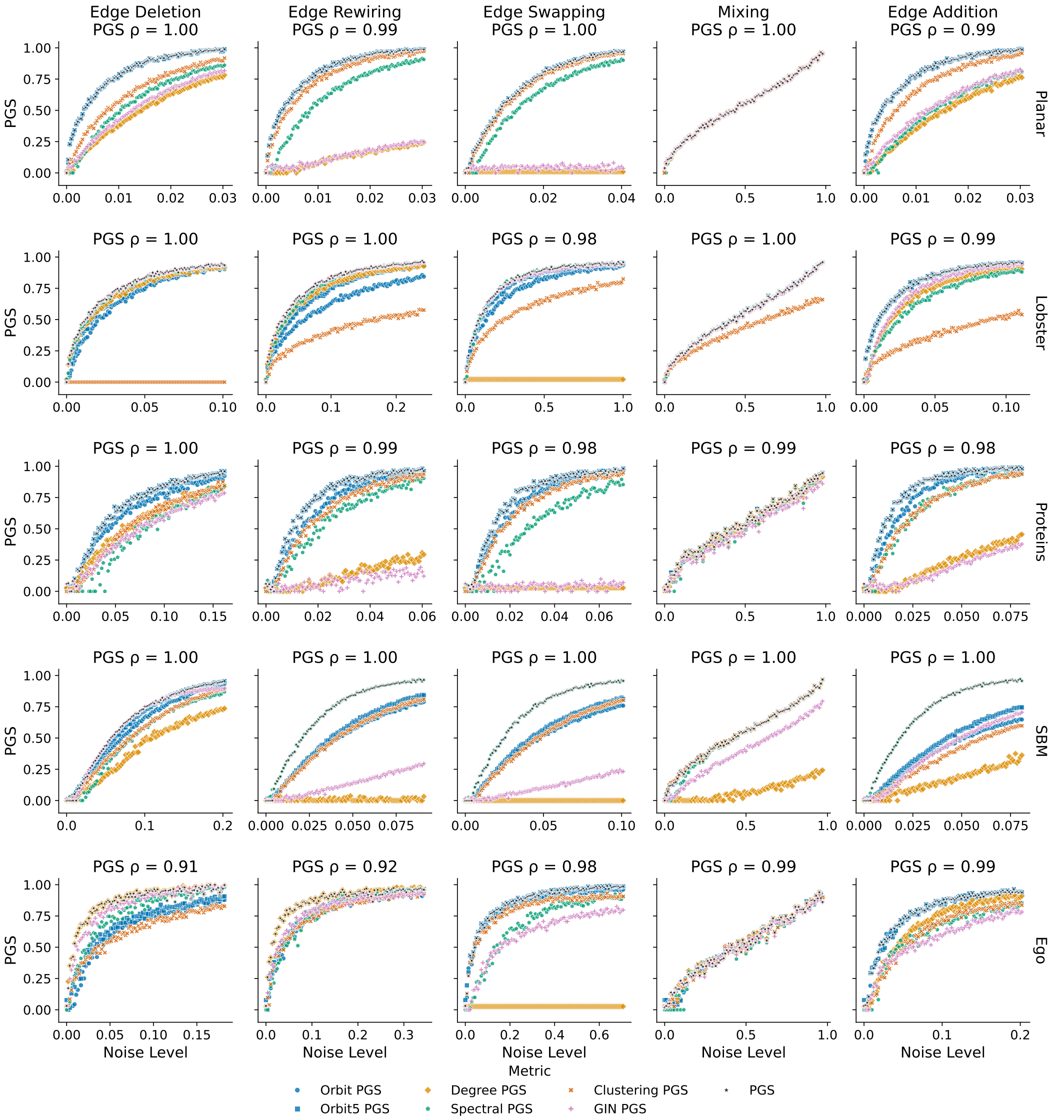


Figure 20: Behavior of descriptor-specific and aggregated PGS (JS) as data distributions are perturbed. The perturbation type varies across rows, while the dataset varies across columns. The Spearman correlation of the aggregate PGS and the perturbation level is denoted by $\rho$.

# I Supplemental for: PGS Correlates With Model Quality

In this section, we provide further details for the experiments presented in Section 5.3.

In Table 9 we provide the exact MMD metrics attained by DiGress as the number of denoising iterations is varied. Analogously, we provide the values of the PGS and descriptor-specific subscores in Table 10. We find that orbit counts appear to be the most discriminative descriptors, as they lead to the highest PGS values.

Table 9: Behavior of RBF-based MMD metrics as the number of denoising steps in DiGress is varied. A separate model is trained for each row for 5k epochs on Planar-L.

| # Steps | Validity | Orbit RBF | Deg. RBF | Spec. RBF | Clust. RBF | GIN RBF |
|---|---|---|---|---|---|---|
| 15 | 0.00 | 0.6460 | 0.0751 | 0.0305 | 0.4751 | 0.2041 |
| 30 | 4.05 | 0.1879 | 0.0280 | 0.0090 | 0.1206 | 0.0956 |
| 45 | 18.70 | 0.0921 | 0.0208 | 0.0049 | 0.0584 | 0.0660 |
| 60 | 30.76 | 0.0680 | 0.0159 | 0.0034 | 0.0377 | 0.0468 |
| 75 | 44.09 | 0.0506 | 0.0182 | 0.0028 | 0.0349 | 0.0350 |
| 90 | 51.27 | 0.0432 | 0.0158 | 0.0025 | 0.0258 | 0.0321 |

Table 10: Behavior of PGS as the number of denoising steps in DiGress is varied. A separate model is trained for each row for 5k epochs on Planar-L.

| # Steps | Validity | PGS | Orbit PGS | Orbit5 PGS | Deg. PGS | Spec. PGS | Clust. PGS | GIN PGS |
|---|---|---|---|---|---|---|---|---|
| 15 | 0.00 | 99.96 | 99.99 | **99.96** | 68.74 | 99.25 | 99.94 | 78.65 |
| 30 | 4.05 | 96.76 | 96.84 | **96.76** | 43.14 | 80.15 | 89.89 | 57.04 |
| 45 | 18.70 | 90.48 | 89.84 | **90.48** | 33.34 | 66.30 | 75.75 | 44.99 |
| 60 | 30.76 | 84.03 | 82.49 | **84.03** | 29.09 | 52.39 | 67.34 | 39.07 |
| 75 | 44.09 | 74.90 | 73.35 | **74.90** | 32.69 | 45.69 | 57.50 | 38.75 |
| 90 | 51.27 | 69.16 | 67.13 | **69.16** | 28.11 | 41.43 | 48.94 | 35.08 |

In Fig. 21, we supplement the experiments presented previously in Fig. 5 with the corresponding results on Planar-L and Lobster-L.

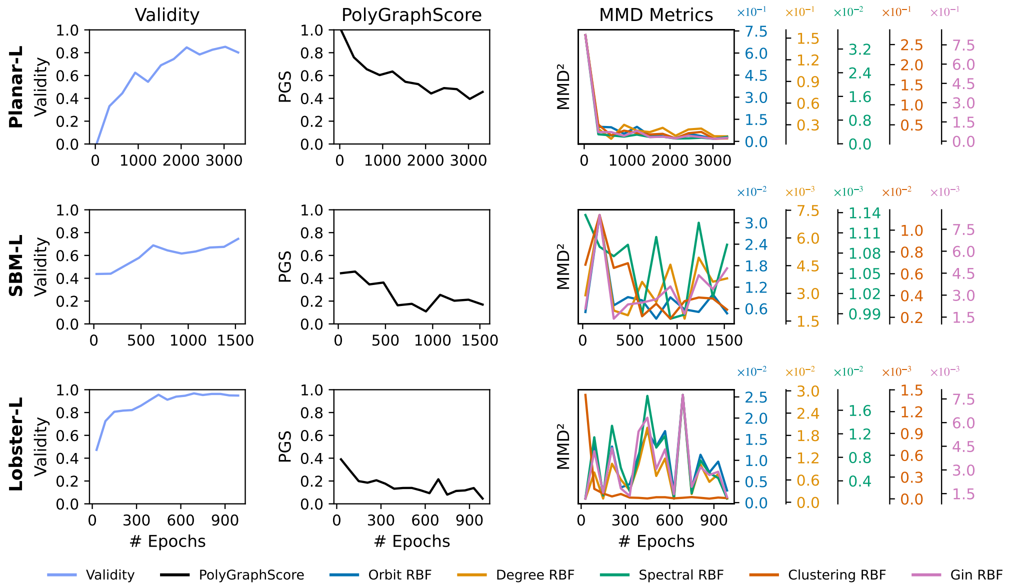


Figure 21: Behavior of validity, PGS, and MMD metrics throughout training of DiGress on procedural datasets.

# J Ablation: TabPFN vs Logistic Regression

In this section, we study logistic regression as an alternative to TabPFN as a discriminator. To this end, we repeat the perturbation experiments from Section 5.2 and Appendix H with logistic regression as a discriminator. We refer to the PGS variant using logistic regression LR PGS.

In Fig. 22 we plot the response of PGS and LR PGS to synthetic perturbations. We find that TabPFN consistently produces PGS estimates that are at least as high as those obtained by logistic regression. In some cases, TabPFN clearly outperforms logistic regression. This may be attributed to the fact that TabPFN can model non-linear decision boundaries and is thus more powerful than logistic regression. We also qualitatively observe that logistic regression leads to a noisier response to the variation of perturbation magnitude.

Hence, since TabPFN simultaneously produces tighter bounds and less noisy estimates, we prefer it to logistic regression.

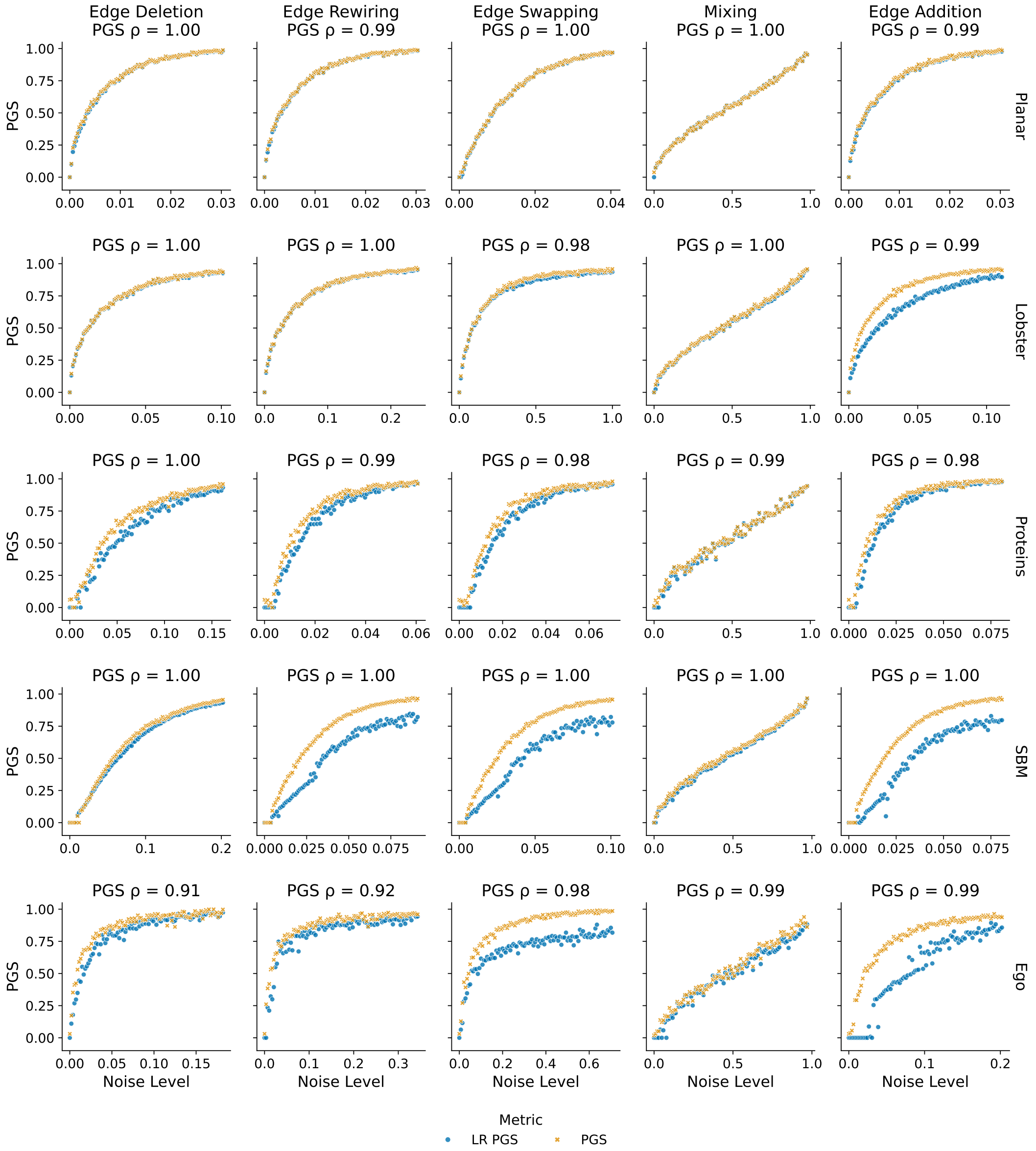


Figure 22: Comparing the behavior of the aggregated PGS (JS) computed via logistic regression (LR PGS) to the aggregated PGS computed via a TabPFN classifier (PGS).

# K Supplemental for: Benchmarking Representative Models

In this Appendix, we do a thorough benchmark of PGS and MMD on Lobster-L, Planar-L, SBM-L and Proteins. To obtain the standard deviations for PGS scores and MMD values in Tables 4 and 11 to 13, we subsample half of the dataset *without replacement* (2048 samples for procedural datasets, and 92 samples for proteins) 10 times. In all those tables, means and standard deviations are scaled by a factor of 100 for legibility purposes. The time taken to compute each of those metrics is reported in Table 15. Timing experiments were run on a compute node equipped with two AMD EPYC 9534 CPUs (using 10 vCPUs in total), an NVIDIA H100 GPU with 80 GB memory (CUDA 12.2, driver 535.230.02), and 128 GB system RAM. Reference values (i.e. the score obtained by computing the metric between the train and test set) for all metrics discussed are in Tables 16 and 17. We note that the PGS discrepancy between the train and test set of MOSES is relatively high, as the test set consists of a separate scaffold split. Importantly, the PGS between the train and test set is very close to 0 (save for MOSES due to changes in the underlying distribution), further showing the absolute nature of PGS, making it much easier to interpret compared to MMD.

Table 11: Comparison of VUN and PGS with biased Gaussian TV-based MMD formulations from Liao et al. (2019). We computed the standard deviation from 10 subsamples of size 2048 except for Proteins, where the subsample size is 92 (50% of the size of the test set). All MMD hyperparameter choices are specified in table 14.

| **Dataset** | **Model** | **VUN (↑)** | **PGS (↓)** | **GTV MMD$^2$ Deg. (↓)** | **GTV MMD$^2$ Clust. (↓)** | **GTV MMD$^2$ Orb. (↓)** | **GTV MMD$^2$ Eig. (↓)** |
|---|---|---|---|---|---|---|---|
| Planar-L | AutoGraph | 0.851 | **33.965** ± 1.786 | 7.814e-05 ± 2.508e−05 | **1.630e-03** ± 2.971e−04 | **1.088e-04** ± 3.000e−05 | 8.229e-04 ± 4.737e−05 |
| | DiGress | 0.801 | 45.189 ± 1.770 | 6.317e-04 ± 4.638e−05 | 1.438e-02 ± 1.203e−03 | 3.675e-03 ± 5.031e−04 | 1.284e-03 ± 5.673e−05 |
| | GRAN | 0.016 | 99.663 ± 0.171 | 6.272e-05 ± 1.422e−05 | 4.658e-03 ± 6.998e−04 | 6.620e-04 ± 1.940e−04 | 1.198e-03 ± 8.786e−05 |
| | ESGG | **0.939** | 45.010 ± 1.395 | **2.288e-05** ± 9.304e−06 | 4.196e-03 ± 5.231e−04 | 1.466e-03 ± 3.001e−04 | **6.797e-04** ± 4.239e−05 |
| Lobster-L | AutoGraph | 0.831 | 18.022 ± 1.608 | 4.453e-04 ± 5.697e−05 | 3.336e-06 ± 1.710e−06 | 6.333e-03 ± 8.331e−04 | 9.893e-04 ± 1.120e−04 |
| | DiGress | **0.914** | **3.167** ± 2.607 | **2.816e-05** ± 1.870e−05 | 1.067e-06 ± 6.954e−07 | **4.166e-04** ± 2.512e−04 | **1.571e-04** ± 1.937e−05 |
| | GRAN | 0.413 | 85.370 ± 0.501 | 1.158e-02 ± 4.212e−04 | 3.344e-03 ± 2.449e−04 | 1.955e-01 ± 6.405e−03 | 2.303e-02 ± 5.802e−04 |
| | ESGG | 0.709 | 69.886 ± 0.557 | 6.252e-03 ± 3.830e−04 | **0.000e+00** ± 0.000e+00 | 6.359e-02 ± 2.028e−03 | 1.030e-02 ± 4.646e−04 |
| SBM-L | AutoGraph | **0.856** | **5.638** ± 1.455 | **4.897e-05** ± 1.537e−05 | **1.017e-03** ± 2.634e−05 | **1.080e-03** ± 2.241e−04 | **1.400e-04** ± 1.859e−05 |
| | DiGress | 0.730 | 17.384 ± 2.285 | 7.500e-04 ± 1.785e−04 | 1.048e-03 ± 2.796e−05 | 2.307e-03 ± 3.480e−04 | 2.449e-04 ± 4.943e−05 |
| | GRAN | 0.214 | 69.114 ± 1.445 | 9.540e-03 ± 3.929e−04 | 3.040e-03 ± 7.289e−05 | 1.306e-02 ± 7.980e−04 | 1.104e-03 ± 7.706e−05 |
| | ESGG | 0.104 | 99.374 ± 0.212 | 3.482e-03 ± 2.877e−04 | 5.687e-03 ± 1.007e−04 | 4.546e-02 ± 1.449e−03 | 2.736e-02 ± 3.318e−04 |
| Proteins | AutoGraph | - | **67.661** ± 7.409 | 2.454e-03 ± 6.456e−04 | 3.750e-02 ± 4.022e−03 | 1.759e-02 ± 3.502e−03 | 2.708e-03 ± 1.730e−04 |
| | DiGress | - | 88.118 ± 3.075 | **2.039e-04** ± 8.748e−05 | **2.471e-02** ± 3.015e−03 | 2.263e-02 ± 7.034e−03 | **1.073e-03** ± 5.723e−05 |
| | GRAN | - | 89.674 ± 2.687 | 3.286e-02 ± 1.852e−03 | 1.068e-01 ± 4.791e−03 | 2.841e-01 ± 1.214e−02 | 9.344e-03 ± 5.235e−04 |
| | ESGG | - | 79.238 ± 4.254 | 1.518e-03 ± 2.904e−04 | 4.031e-02 ± 1.987e−03 | **6.474e-03** ± 1.315e−03 | 1.269e-03 ± 1.318e−04 |

Table 12: Unbiased RBF kernel-based MMD estimates. We computed the standard deviation from 10 subsamples of size 2048 except for Proteins, where the subsample size is 92 (50% of the size of the test set). All MMD hyperparameter choices are specified in table 14.

| **Dataset** | **Model** | **VUN (↑)** | **PGS (↓)** | **RBF MMD$^2$ Deg. (↓)** | **RBF MMD$^2$ Clust. (↓)** | **RBF MMD$^2$ Orb. (↓)** | **RBF MMD$^2$ Eig. (↓)** |
|---|---|---|---|---|---|---|---|
| Planar-L | AutoGraph | 0.851 | **33.965** ± 1.786 | 1.961e-03 ± 6.688e−04 | **5.616e-04** ± 1.687e−04 | **2.488e-03** ± 3.395e−04 | 1.035e-03 ± 7.394e−05 |
| | DiGress | 0.801 | 45.189 ± 1.770 | 1.623e-02 ± 1.130e−03 | 1.487e-02 ± 1.508e−03 | 3.059e-02 ± 3.484e−03 | 1.713e-03 ± 9.201e−05 |
| | GRAN | 0.016 | 99.663 ± 0.171 | 3.250e-03 ± 6.760e−04 | 3.761e-03 ± 8.004e−04 | 9.068e-03 ± 7.194e−04 | 4.742e-03 ± 1.555e−04 |
| | ESGG | **0.939** | 45.010 ± 1.395 | **1.322e-03** ± 2.961e−04 | 3.778e-03 ± 6.875e−04 | 2.708e-02 ± 1.779e−03 | **8.337e-04** ± 7.146e−05 |
| Lobster-L | AutoGraph | 0.831 | 18.022 ± 1.608 | 8.446e-03 ± 1.241e−03 | 5.017e-06 ± 2.677e−06 | 7.725e-03 ± 1.340e−03 | 6.748e-03 ± 1.198e−03 |
| | DiGress | **0.914** | **3.167** ± 2.607 | **2.969e-04** ± 5.087e−04 | 1.208e-06 ± 9.237e−07 | **7.208e-04** ± 5.402e−04 | **2.389e-04** ± 2.738e−04 |
| | GRAN | 0.413 | 85.370 ± 0.501 | 2.965e-01 ± 8.501e−03 | 4.605e-03 ± 3.158e−04 | 1.526e-01 ± 4.294e−03 | 1.774e-01 ± 7.080e−03 |
| | ESGG | 0.709 | 69.886 ± 0.557 | 8.650e-02 ± 3.577e−03 | **0.000e+00** ± 0.000e+00 | 2.163e-01 ± 7.297e−03 | 4.552e-02 ± 1.243e−03 |
| SBM-L | AutoGraph | **0.856** | **5.638** ± 1.455 | **2.085e-04** ± 1.663e−04 | **3.275e-04** ± 1.506e−04 | **9.928e-05** ± 6.512e−05 | **7.888e-05** ± 2.978e−05 |
| | DiGress | 0.730 | 17.384 ± 2.285 | 3.385e-03 ± 8.299e−04 | 1.738e-03 ± 3.772e−04 | 4.252e-04 ± 8.053e−05 | 2.832e-04 ± 7.796e−05 |
| | GRAN | 0.214 | 69.114 ± 1.445 | 4.543e-02 ± 1.560e−03 | 4.111e-02 ± 1.828e−03 | 3.194e-03 ± 2.032e−04 | 2.671e-03 ± 2.659e−04 |
| | ESGG | 0.104 | 99.374 ± 0.212 | 3.255e-02 ± 2.096e−03 | 5.523e-02 ± 1.585e−03 | 1.334e-02 ± 2.608e−04 | 2.262e-02 ± 5.563e−04 |
| Proteins | AutoGraph | - | **67.661** ± 7.409 | 4.025e-02 ± 5.459e−03 | 5.165e-02 ± 5.930e−03 | **1.715e-02** ± 2.728e−03 | 3.967e-03 ± 3.339e−04 |
| | DiGress | - | 88.118 ± 3.075 | **2.889e-02** ± 4.234e−03 | **2.230e-02** ± 3.158e−03 | 5.588e-02 ± 1.390e−02 | **1.239e-03** ± 1.592e−04 |
| | GRAN | - | 89.674 ± 2.687 | 2.853e-01 ± 1.816e−02 | 2.495e-01 ± 1.232e−02 | 3.731e-01 ± 1.399e−02 | 2.967e-02 ± 2.078e−03 |
| | ESGG | - | 79.238 ± 4.254 | 5.391e-02 ± 7.314e−03 | 5.968e-02 ± 3.388e−03 | 3.669e-02 ± 8.273e−03 | 1.431e-03 ± 3.791e−04 |

Table 13: Biased RBF kernel-based MMD estimates. We computed the standard deviation from 10 subsamples of size 2048 except for Proteins, where the subsample size is 92 (50% of the size of the test set). All MMD hyperparameter choices are specified in table 14.

| **Dataset** | **Model** | **VUN (↑)** | **PGS (↓)** | **RBF MMD$^2$ Deg. (↓)** | **RBF MMD$^2$ Clust. (↓)** | **RBF MMD$^2$ Orb. (↓)** | **RBF MMD$^2$ Eig. (↓)** |
|---|---|---|---|---|---|---|---|
| PLANAR-L | AutoGraph | 0.851 | **33.965** ± 1.786 | 2.514e-03 ± 6.689e−04 | **1.147e-03** ± 1.681e−04 | **3.154e-03** ± 3.387e−04 | 1.624e-03 ± 7.366e−05 |
| | DIGRESS | 0.801 | 45.189 ± 1.770 | 1.679e-02 ± 1.129e−03 | 1.546e-02 ± 1.507e−03 | 3.098e-02 ± 3.389e−03 | 2.303e-03 ± 9.193e−05 |
| | GRAN | 0.016 | 99.663 ± 0.171 | 3.800e-03 ± 6.768e−04 | 4.347e-03 ± 8.007e−04 | 9.981e-03 ± 6.747e−04 | 5.330e-03 ± 1.557e−04 |
| | ESGG | **0.939** | 45.010 ± 1.395 | **1.884e-03** ± 2.951e−04 | 4.367e-03 ± 6.874e−04 | 2.769e-02 ± 1.831e−03 | **1.423e-03** ± 7.155e−05 |
| LOBSTER-L | AutoGraph | 0.831 | 18.022 ± 1.608 | 9.012e-03 ± 1.239e−03 | 6.324e-06 ± 3.509e−06 | 8.229e-03 ± 1.340e−03 | 7.486e-03 ± 1.197e−03 |
| | DIGRESS | **0.914** | **3.167** ± 2.607 | **8.316e-04** ± 5.338e−04 | 1.760e-06 ± 1.137e−06 | **1.509e-03** ± 4.723e−04 | **9.372e-04** ± 3.026e−04 |
| | GRAN | 0.413 | 85.370 ± 0.501 | 2.972e-01 ± 8.498e−03 | 4.795e-03 ± 3.334e−04 | 1.533e-01 ± 4.293e−03 | 1.782e-01 ± 7.078e−03 |
| | ESGG | 0.709 | 69.886 ± 0.557 | 8.710e-02 ± 3.578e−03 | **0.000e+00** ± 0.000e+00 | 2.167e-01 ± 7.310e−03 | 4.627e-02 ± 1.244e−03 |
| SBM-L | AutoGraph | **0.856** | **5.638** ± 1.455 | **9.239e-04** ± 1.680e−04 | **7.998e-04** ± 7.636e−05 | **1.068e-03** ± 6.223e−05 | **5.036e-04** ± 3.006e−05 |
| | DIGRESS | 0.730 | 17.384 ± 2.285 | 4.100e-03 ± 8.285e−04 | 2.140e-03 ± 2.579e−04 | 1.392e-03 ± 8.057e−05 | 7.082e-04 ± 7.822e−05 |
| | GRAN | 0.214 | 69.114 ± 1.445 | 4.617e-02 ± 1.560e−03 | 3.392e-02 ± 1.390e−03 | 4.163e-03 ± 2.031e−04 | 3.115e-03 ± 2.666e−04 |
| | ESGG | 0.104 | 99.374 ± 0.212 | 3.329e-02 ± 2.095e−03 | 3.564e-02 ± 9.014e−04 | 1.430e-02 ± 2.607e−04 | 2.313e-02 ± 5.551e−04 |
| Proteins | AutoGraph | - | **67.661** ± 7.409 | 4.648e-02 ± 5.412e−03 | 5.857e-02 ± 5.924e−03 | **2.674e-02** ± 2.481e−03 | 6.070e-03 ± 3.329e−04 |
| | DIGRESS | - | 88.118 ± 3.075 | **3.500e-02** ± 4.196e−03 | **2.876e-02** ± 3.076e−03 | 6.312e-02 ± 1.386e−02 | **3.605e-03** ± 1.646e−04 |
| | GRAN | - | 89.674 ± 2.687 | 2.917e-01 ± 1.812e−02 | 2.543e-01 ± 1.237e−02 | 3.784e-01 ± 1.398e−02 | 3.228e-02 ± 2.108e−03 |
| | ESGG | - | 79.238 ± 4.254 | 6.034e-02 ± 7.284e−03 | 6.691e-02 ± 3.333e−03 | 4.505e-02 ± 8.287e−03 | 3.923e-03 ± 4.190e−04 |

Table 14: Mapping of display columns in results tables to MMD configurations. For all RBF MMDs, the final MMD was computed as the maximum value over the following bandwidths $\{\sigma_i\}_{i=1}^6 = \{0.1, 0.5, 1.0, 2.0, 5.0, 10.0\}$ as per Thompson et al. (2022). For the descriptor parameters, we used 100,000 for the width of the sparse degree histogram, 100 bins for the clustering histogram, and 4 for the orbit count. RBF: radial basis function; GTV: Gaussian total variation distance; UMVE: unbiased minimum variance estimator, see Gretton et al. (2012).

| **Name** | **Variant** | **Kernel** | | **Descriptor** |
|---|---|---|---|---|
| | | **Name** | **Parameter** | |
| GTV MMD$^2$ Deg. | Biased | GTV | 1.0 | Degree |
| GTV MMD$^2$ Clust. | | | 0.1 | Clustering |
| GTV MMD$^2$ Orb. | | | 30 | Orbit |
| GTV MMD$^2$ Eig. | | | 1.0 | Eigenvalues |
| RBF MMD$^2$ Deg. | UMVE | RBF | $\{\sigma_i\}_{i=1}^6$ | Degree |
| RBF MMD$^2$ Clust. | | | | Clustering |
| RBF MMD$^2$ Orb. | | | | Orbit |
| RBF MMD$^2$ Eig. | | | | Eigenvalues |
| RBF MMD$^2$ Deg. | Biased | RBF | $\{\sigma_i\}_{i=1}^6$ | Degree |
| RBF MMD$^2$ Clust. | | | | Clustering |
| RBF MMD$^2$ Orb. | | | | Orbit |
| RBF MMD$^2$ Eig. | | | | Eigenvalues |

Table 15: Compute time (s) per metric across datasets. Standard deviations are obtained from the metrics computed on different model samples. Caching of intermediate or reused MMD values in `PolyGraph` help make MMD computations substantially faster. Int. indicates whether the metric yields an interval through subsampling. VUN scores were parallelized across 10 CPUs.

| Metric | Int. | PLANAR-L | LOBSTER-L | SBM-L | Proteins | Overall |
|---|---|---|---|---|---|---|
| VUN | ✗ | 425.60 ± 17.72 | 253.32 ± 8.95 | 1181.26 ± 101.98 | - | 620.06 ± 37.98 |
| PGS | ✗ | 73.64 ± 3.01 | 338.82 ± 190.27 | 125.02 ± 17.77 | 140.35 ± 73.67 | 169.46 ± 52.81 |
| PGS | ✓ | 192.13 ± 14.27 | 696.11 ± 367.93 | 280.98 ± 47.45 | 223.67 ± 111.39 | 348.22 ± 118.35 |
| RBF MMD$^2$ Deg. | ✓ | 12.61 ± 0.33 | 12.38 ± 0.14 | 12.72 ± 0.27 | 3.68 ± 0.59 | 10.35 ± 0.23 |
| Biased RBF MMD$^2$ Deg. | ✓ | 10.50 ± 0.21 | 10.32 ± 0.24 | 10.46 ± 0.21 | 1.49 ± 0.65 | 8.19 ± 0.23 |
| GTV MMD$^2$ Deg. | ✓ | 7.74 ± 1.22 | 8.04 ± 0.21 | 8.46 ± 2.54 | 3.26 ± 0.40 | 6.88 ± 0.78 |
| GTV MMD$^2$ Deg. | ✗ | 3.53 ± 0.25 | 3.54 ± 0.32 | 3.83 ± 0.39 | 3.51 ± 0.70 | 3.60 ± 0.21 |
| RBF MMD$^2$ Clust. | ✓ | 16.23 ± 0.46 | 13.69 ± 0.38 | 22.63 ± 1.61 | 16.48 ± 8.16 | 17.26 ± 1.86 |
| Biased RBF MMD$^2$ Clust. | ✓ | 16.60 ± 0.50 | 14.00 ± 1.22 | 25.60 ± 1.86 | 16.73 ± 8.25 | 18.23 ± 2.26 |
| GTV MMD$^2$ Clust. | ✓ | 11.80 ± 1.17 | 10.24 ± 0.13 | 16.75 ± 2.00 | 14.16 ± 8.20 | 13.24 ± 1.93 |
| GTV MMD$^2$ Clust. | ✗ | 7.63 ± 0.06 | 5.54 ± 0.13 | 12.90 ± 2.09 | 14.27 ± 8.35 | 10.08 ± 2.03 |
| RBF MMD$^2$ Orb. | ✓ | 11.87 ± 0.20 | 11.84 ± 0.32 | 14.58 ± 0.62 | 4.84 ± 2.64 | 10.78 ± 0.66 |
| Biased RBF MMD$^2$ Orb. | ✓ | 11.82 ± 0.07 | 11.95 ± 0.36 | 14.64 ± 0.52 | 4.75 ± 2.69 | 10.79 ± 0.70 |
| GTV MMD$^2$ Orb. | ✓ | 5.75 ± 1.08 | 5.85 ± 0.08 | 6.71 ± 1.31 | 3.73 ± 2.13 | 5.51 ± 0.56 |
| GTV MMD$^2$ Orb. | ✗ | 1.64 ± 0.02 | 1.22 ± 0.02 | 2.73 ± 0.41 | 3.71 ± 2.12 | 2.32 ± 0.50 |
| RBF MMD$^2$ Eig. | ✓ | 21.56 ± 0.83 | 19.13 ± 0.71 | 25.83 ± 1.47 | 31.99 ± 16.42 | 24.63 ± 4.14 |
| Biased RBF MMD$^2$ Eig. | ✓ | 25.16 ± 6.52 | 18.75 ± 0.47 | 25.86 ± 1.84 | 33.11 ± 16.31 | 25.72 ± 2.80 |
| GTV MMD$^2$ Eig. | ✓ | 17.85 ± 1.18 | 17.55 ± 0.24 | 20.77 ± 1.83 | 29.67 ± 17.44 | 21.46 ± 4.21 |
| GTV MMD$^2$ Eig. | ✗ | 13.80 ± 0.09 | 12.92 ± 0.16 | 16.88 ± 1.56 | 32.26 ± 19.52 | 18.97 ± 4.82 |

Table 16: Reference values between the test and training set for various metrics.

| **Metric** | PLANAR-L | LOBSTER-L | SBM-L | Proteins |
|---|---|---|---|---|
| **PGS (↓)** | 0.6 ± 1.2 | 0.8 ± 1.6 | 0.2 ± 0.6 | 2.1 ± 3.4 |
| **Clust. (↓)** | 0.1 ± 0.4 | 0.0 ± 0.0 | 0.1 ± 0.2 | 3.2 ± 3.6 |
| **Deg. (↓)** | 1.4 ± 1.4 | 0.7 ± 1.1 | 0.6 ± 1.1 | 5.2 ± 3.9 |
| **GIN (↓)** | 0.1 ± 0.4 | 0.4 ± 0.8 | 0.1 ± 0.4 | 3.0 ± 3.2 |
| **Orb5. (↓)** | 0.2 ± 0.5 | 0.5 ± 0.8 | 0.0 ± 0.1 | 1.1 ± 2.0 |
| **Orb4. (↓)** | 0.5 ± 0.7 | 0.6 ± 1.1 | 0.3 ± 0.6 | 2.0 ± 2.5 |
| **Eig. (↓)** | 0.0 ± 0.0 | 1.3 ± 1.5 | 0.2 ± 0.6 | 0.9 ± 2.7 |
| **GTV MMD$^2$ Clust. (↓)** | 2.91e-04 | 0.00e+00 | 4.87e-04 | 0.0068 |
| **GTV MMD$^2$ Clust. (↓)** | 5.87e-04 ± 1.3e−04 | 0.00e+00 ± 0.0e+00 | 9.69e-04 ± 9.4e−06 | 0.0104 ± 9.4e−04 |
| **RBF MMD$^2$ Clust. (↓)** | 3.44e-05 ± 5.1e−05 | 0.00e+00 ± 0.0e+00 | 1.62e-06 ± 3.7e−06 | 0.0014 ± 0.0016 |
| **RBF MMD$^2$ Clust. (↓)** | 5.34e-04 ± 1.5e−04 | 0.00e+00 ± 0.0e+00 | 6.10e-04 ± 2.6e−05 | 0.0077 ± 0.0020 |
| **GTV MMD$^2$ Deg. (↓)** | 1.51e-05 | 1.79e-05 | 1.69e-05 | 3.16e-04 |
| **GTV MMD$^2$ Deg. (↓)** | 2.14e-05 ± 1.1e−05 | 3.06e-05 ± 1.3e−05 | 3.86e-05 ± 2.4e−05 | 5.67e-04 ± 4.6e−04 |
| **RBF MMD$^2$ Deg. (↓)** | 1.69e-04 ± 1.7e−04 | 1.19e-04 ± 1.2e−04 | 1.48e-04 ± 1.2e−04 | 0.0052 ± 0.0038 |
| **RBF MMD$^2$ Deg. (↓)** | 6.38e-04 ± 2.7e−04 | 6.03e-04 ± 2.0e−04 | 8.54e-04 ± 1.3e−04 | 0.0117 ± 0.0039 |
| **GTV MMD$^2$ Orb. (↓)** | 3.43e-06 | 1.36e-05 | 3.26e-04 | 0.0032 |
| **GTV MMD$^2$ Orb. (↓)** | 2.18e-05 ± 2.1e−05 | 5.79e-05 ± 2.8e−05 | 8.79e-04 ± 2.1e−04 | 0.0065 ± 0.0042 |
| **RBF MMD$^2$ Orb. (↓)** | 1.05e-04 ± 9.8e−05 | 3.41e-04 ± 2.8e−04 | 2.98e-05 ± 3.7e−05 | 0.0044 ± 0.0055 |
| **RBF MMD$^2$ Orb. (↓)** | 0.0010 ± 3.3e−05 | 0.0012 ± 2.3e−04 | 9.99e-04 ± 3.4e−05 | 0.0132 ± 0.0038 |
| **GTV MMD$^2$ Eig. (↓)** | 7.39e-05 | 5.12e-05 | 4.93e-05 | 4.85e-04 |
| **GTV MMD$^2$ Eig. (↓)** | 1.27e-04 ± 2.5e−05 | 1.10e-04 ± 2.6e−05 | 9.75e-05 ± 1.9e−05 | 6.97e-04 ± 1.1e−04 |
| **RBF MMD$^2$ Eig. (↓)** | 1.69e-05 ± 2.9e−05 | 2.78e-05 ± 4.0e−05 | 5.21e-06 ± 9.7e−06 | 1.41e-04 ± 2.1e−04 |
| **RBF MMD$^2$ Eig. (↓)** | 5.80e-04 ± 5.0e−05 | 6.43e-04 ± 1.0e−04 | 4.02e-04 ± 3.1e−05 | 0.0024 ± 2.9e−04 |

Table 17: Reference PGS metrics between the molecule test and training sets. Note that MOSES uses a scaffold split, resulting in a high discrepancy between the train and test set.

| **Dataset** | | **PGS subscores** | | | | |
|---|---|---|---|---|---|---|
| | **PGS (↓)** | **Topo (↓)** | **Morgan (↓)** | **ChemNet (↓)** | **MolCLR (↓)** | **Lipinski (↓)** |
| GUACAMOL | 0.2 ± 0.4 | 0.2 ± 0.4 | 0.3 ± 0.5 | 0.3 ± 0.6 | 0.1 ± 0.2 | 0.0 ± 0.0 |
| MOSES | 21.0 ± 0.6 | 21.0 ± 0.6 | 17.8 ± 0.7 | 16.0 ± 1.2 | 18.0 ± 0.8 | 20.7 ± 0.7 |

# L STABILITY OF PGS UNDER VARYING SAMPLE SIZES.

Figs. 23 to 26 show the relationship between the PGS score and the number of samples. The PGS score of the reference graphs with respect to another set of reference graphs issued from the same distribution is given as a comparison. For all experiments, we show the mean as well as the 5th and 95th quantile to give an estimate of the variance of PGS at different sample sizes.

For most models, some separation from the test set occurs above 256 samples, with PGS scores, and especially the upper bound is mostly stable beyond this range. This both showcases the stability of the metric, the number of samples required to get a reliable PGS estimate, as well as the overall PGS ranges for the various models we considered for this study.

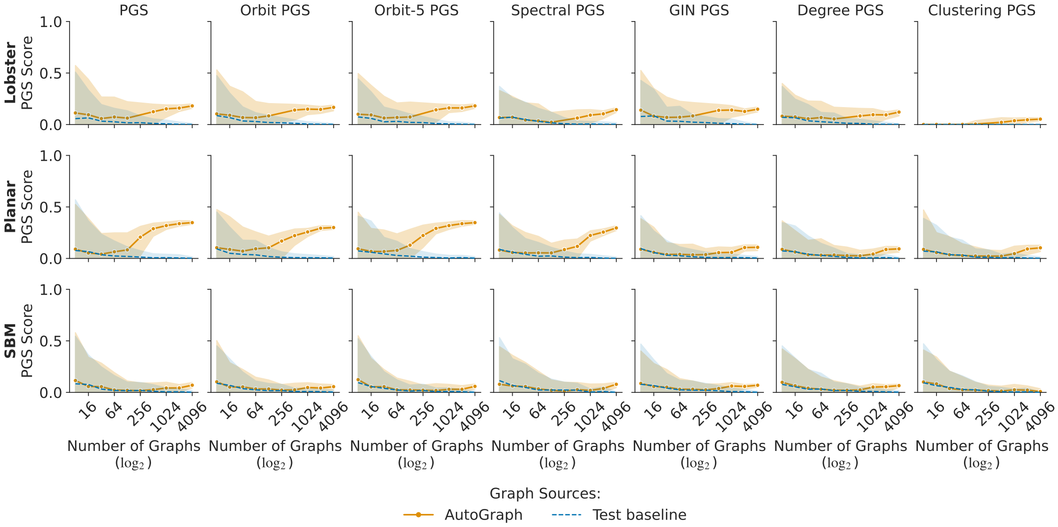


Figure 23: PGS obtained from varying sample sizes generated by AutoGraph.

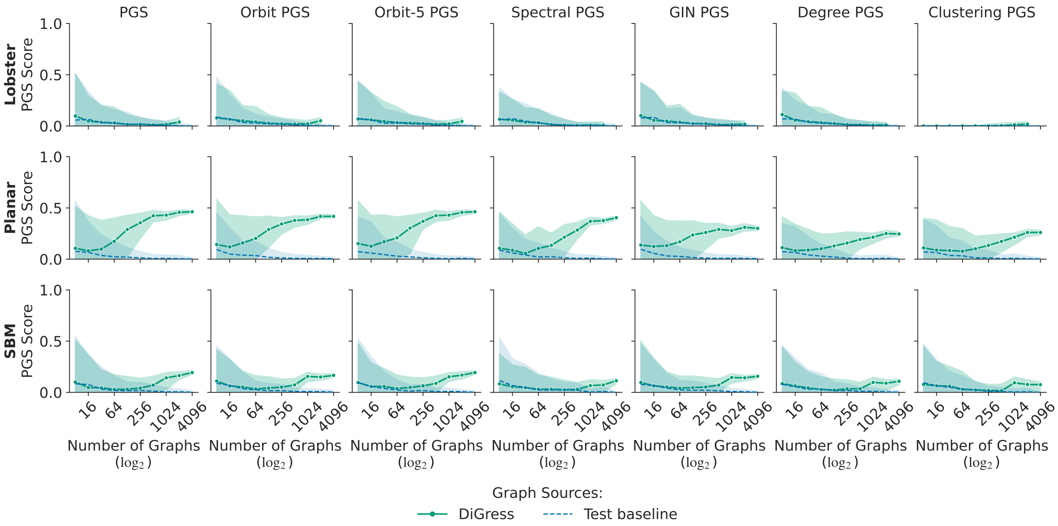


Figure 24: PGS obtained from varying sample sizes generated by DIGRESS.

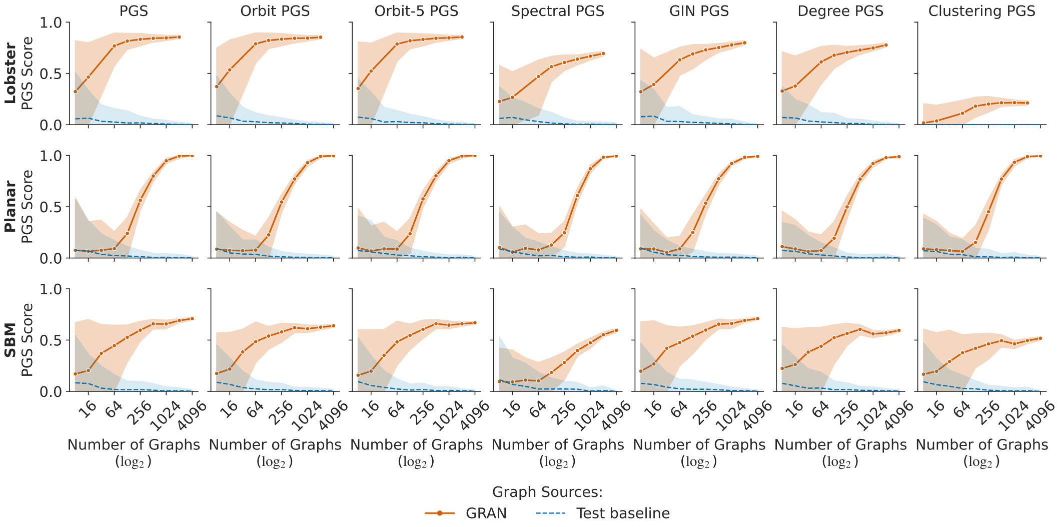


Figure 25: PGS obtained from varying sample sizes generated by GRAN.

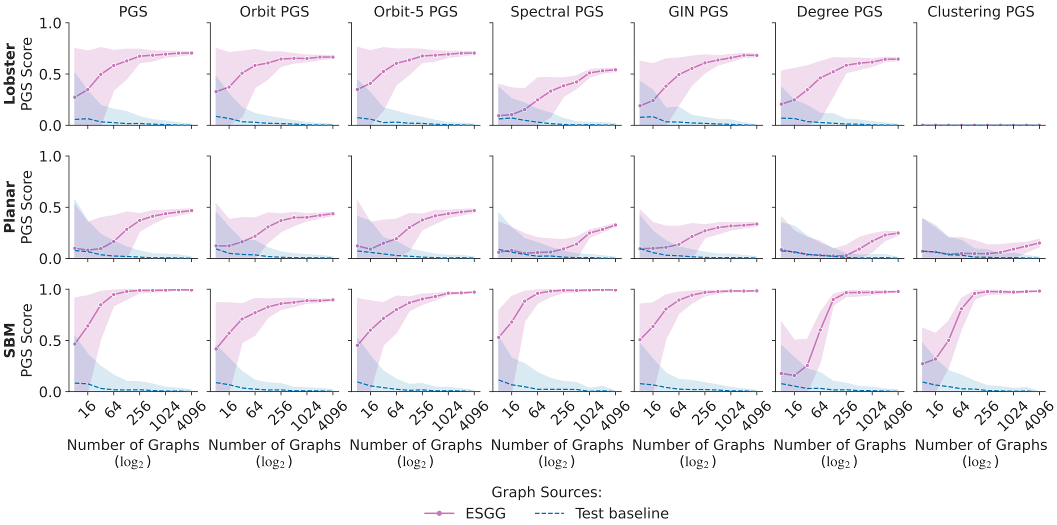


Figure 26: PGS obtained from varying sample sizes generated by ESGG.

# M Larger Procedural Reference Datasets for Better GGM Benchmarking

Following our findings of Section 5.1 and Appendix G, we introduce larger procedurally-generated datasets for planar, lobster and SBM graphs, which we term Planar-L, Lobster-L and SBM-L. Lobster-L is a set of tree-shaped lobster graphs generated using `nx.random_lobster`, controlled by expected node count (80) and attachment probabilities to the backbone and its neighbors (set to $0.7$ for both). Planar-L is a set of connected planar graphs generated by uniformly sampling $64$ node positions in the unit square and forming the Delaunay triangulation via `scipy.spatial.Delaunay`, yielding planar edge sets from triangle simplices. SBM-L is a set of stochastic block model graphs with the number of communities sampled uniformly from $2$ to $5$ and nodes per community from $20$ to $40$, where edges are drawn with intra-community probability $0.3$ and inter-community probability $0.005$. SBM-L, Planar-L, and Lobster-L datasets follow `networkx`'s BSD-3 license.

Table 18: Dataset sizes (number of graphs) per split.

| Dataset | Train | Val | Test |
|---|---|---|---|
| SBM-L | 8192 | 4096 | 4096 |
| Planar-L | 8192 | 4096 | 4096 |
| Lobster-L | 8192 | 4096 | 4096 |

# N Influence of Training Set Size on AutoGraph

As shown in Fig. 27, AutoGraph converges to similar VUN values across datasets, yet the loss is substantially lower for SBM-L after training than for SBM-S. This finding indicates that models may overfit on the existing small procedural datasets, further drawing into question the validity of previously reported evaluation results (Vignac et al., 2023).

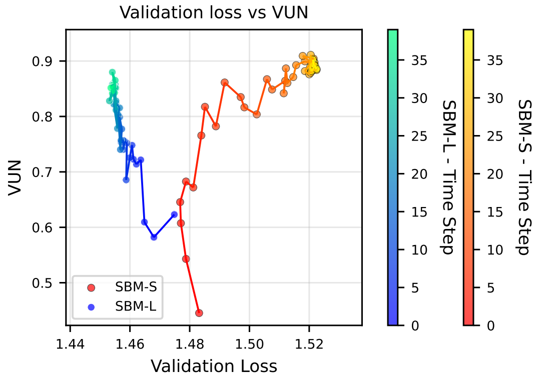


Figure 27: VUN vs. Loss for AutoGraph over the course of a training run.

# O Common Shortfalls of Existing Solutions

To address the lack of inherent scale in MMD, some have proposed normalizing the MMD between generated and test graphs by the MMD between train and test graphs (Martinkus et al., 2022). However, this approach has several shortcomings:

**Limited theoretical justification** MMD was originally introduced as a kernel two-sample test. Its manipulation beyond direct use as a performance metric or for $p$-value computation remains poorly understood.

**Lack of composability** The MMD ratio does not enable combining information across multiple descriptors.

**Sample-size sensitivity** As shown in Appendix G, MMD strongly depends on sample size. Dividing MMDs computed on different sample sizes produces ratios with unclear or unreliable interpretation.

## P Dataset details

Here, we provide details about the datasets used in this study. Licenses for those datasets are summarized in Table 19. Table 20 shows the dataset statistics of the Citeseer dataset (Sen et al., 2008). The statistics for the small procedural datasets are presented in Table 21 (Planar), Table 22 (SBM), and Table 23 (Lobster).

Table 19: License and author information of the datasets used in our experiments.

| **Dataset** | **Author** | **License** |
|---|---|---|
| Citeseer | (Sen et al., 2008) | CC BY-NC-SA 3.0 |
| Procedural (Planar, SBM, Lobster) | (Martinkus et al., 2022; Hagberg et al., 2008) | BSD-3 |
| Proteins | (Dobson & Doig, 2003) | CC0 1.0 Universal |

Table 20: Ego dataset statistics (extracted from Citeseer).

| **Metric** | **Train** | **Val** | **Test** |
|---|---|---|---|
| Number of Graphs | 454 | 151 | 152 |
| Minimum number of Nodes | 50 | 50 | 50 |
| Maximum number of Nodes | 399 | 333 | 364 |
| Average number of Nodes | 141.72 | 139.29 | 158.08 |
| Minimum number of Edges | 64 | 56 | 63 |
| Maximum number of Edges | 1066 | 898 | 1004 |
| Average number of Edges | 325.16 | 321.87 | 369.30 |
| Edge/Node Ratio | 2.29 | 2.31 | 2.34 |

Table 21: Dataset statistics for the Planar dataset (train, validation, and test splits).

| **Metric** | **Train** | **Validation** | **Test** |
|---|---|---|---|
| Number of Graphs | 128 | 32 | 40 |
| Minimum Number of Nodes | 64 | 64 | 64 |
| Maximum Number of Nodes | 64 | 64 | 64 |
| Average Number of Nodes | 64.00 | 64.00 | 64.00 |
| Minimum Number of Edges | 173 | 174 | 174 |
| Maximum Number of Edges | 181 | 181 | 181 |
| Average Number of Edges | 177.83 | 177.75 | 177.93 |
| Edge-to-Node Ratio | 2.78 | 2.78 | 2.78 |

## Q Feature concatenation as an alternative to max-reduction

An alternative to taking the maximum JSD across descriptors consists of obtaining an overall PGS score by concatenating all vectors arising from the different descriptors, and training a discriminator atop these concatenated features. However, because we are working with TabPFN and want to keep the discriminator computation time reasonable, we need to apply a dimensionality reduction technique (here, we choose PCA) for this concatenated vector to fit within the feature limits recommended by TabPFN (for v2.0, this is 500). This makes attributing potentially high values to specific descriptors impossible, but in practice still results in a tighter bound (i.e., higher scores) as can be seen in Table 24.

Table 22: Dataset statistics for the SBM dataset (train, validation, and test splits).

| **Metric** | **Train** | **Validation** | **Test** |
|---|---|---|---|
| Number of Graphs | 128 | 32 | 40 |
| Minimum Number of Nodes | 44 | 49 | 54 |
| Maximum Number of Nodes | 187 | 162 | 174 |
| Average Number of Nodes | 105.99 | 91.28 | 107.85 |
| Minimum Number of Edges | 129 | 183 | 210 |
| Maximum Number of Edges | 1129 | 857 | 972 |
| Average Number of Edges | 512.51 | 425.19 | 521.88 |
| Edge-to-Node Ratio | 4.84 | 4.66 | 4.84 |

Table 23: Dataset statistics for the Lobster dataset (train, validation, and test splits).

| **Metric** | **Train** | **Validation** | **Test** |
|---|---|---|---|
| Number of Graphs | 60 | 20 | 20 |
| Minimum Number of Nodes | 10 | 11 | 14 |
| Maximum Number of Nodes | 98 | 98 | 84 |
| Average Number of Nodes | 53.67 | 56.30 | 50.80 |
| Minimum Number of Edges | 9 | 10 | 13 |
| Maximum Number of Edges | 97 | 97 | 83 |
| Average Number of Edges | 52.67 | 55.30 | 49.80 |
| Edge-to-Node Ratio | 0.98 | 0.98 | 0.98 |

## R KERNEL LOGISTIC REGRESSION WITH GRAPH KERNELS

One can adopt the kernel logistic regression classifier and use graph kernels directly to evaluate GGMs, effectively showing that any (graph) kernel also suitable for MMD can also be used in PGS. We showcase this with the Weisfeiler-Lehman (Shervashidze et al., 2011), shortest path (Borgwardt & Kriegel, 2005), and PyramidMatch (Grauman & Darrell, 2007) graph kernels in Table 25. However, they almost always show looser bounds compared to the standard PGS formulation, so we do not favor such kernels.

## S USE OF LARGE LANGUAGE MODELS

The authors used large language models in the following ways:

**Intelligent tab completion** During software development, tools for intelligent line-wise tab completion were used.

**Preparation of visualizations** LLMs were partly used to generate code for figure layouts. The correctness of all code and data was checked manually. The data shown in the figures was generated by manually written code.

**Information retrieval** LLMs were queried for related work, but produced no relevant results. All related work presented in the manuscript was manually retrieved, save for Endres & Schindelin (2003), which was manually checked to contain the required proof.

**Polishing of manuscript** LLMs were occasionally used to refine or rephrase individual sentences.

Table 24: Comparison of VUN, max-reduced PGS (the default we also use in Table 4) and PGS with concatenated descriptors. PGS-Concat. is obtained by concatenating all descriptor features, and subsequently applying a dimensionality reduction technique (PCA) for the feature vectors to fit within TabPFN's recommended feature size limit (for v2.0, this is 500). The final score is obtained similarly to PGS.

| **Dataset** | **Model** | **VUN (↑)** | **PGS (↓)** | **PGS-Concat. (↓)** |
|---|---|---|---|---|
| Planar-L | AutoGraph | 85.1 | **34.0** ± 1.8 | **44.8** ± 1.3 |
| | DiGress | 80.1 | 45.2 ± 1.8 | 55.3 ± 1.5 |
| | GRAN | 1.6 | 99.7 ± 0.2 | 99.4 ± 0.2 |
| | ESGG | **93.9** | 45.0 ± 1.4 | 52.4 ± 1.1 |
| Lobster-L | AutoGraph | 83.1 | 18.0 ± 1.6 | **29.0** ± 2.1 |
| | DiGress | **91.4** | **3.2** ± 2.6 | 43.2 ± 1.4 |
| | GRAN | 41.3 | 85.4 ± 0.5 | 86.4 ± 0.9 |
| | ESGG | 70.9 | 69.9 ± 0.6 | 69.9 ± 1.0 |
| SBM-L | AutoGraph | **85.6** | **5.6** ± 1.5 | **27.2** ± 3.0 |
| | DiGress | 72.8 | 17.4 ± 2.3 | 32.0 ± 2.0 |
| | GRAN | 21.4 | 69.1 ± 1.4 | 78.0 ± 0.8 |
| | ESGG | 10.6 | 99.4 ± 0.2 | 98.1 ± 0.4 |
| Proteins | AutoGraph | - | **67.7** ± 7.4 | **94.8** ± 2.6 |
| | DiGress | - | 88.1 ± 3.1 | 99.6 ± 0.3 |
| | GRAN | - | 89.7 ± 2.7 | 99.8 ± 0.1 |
| | ESGG | - | 79.2 ± 4.3 | 99.4 ± 0.3 |

Table 25: Comparison of PGS (as shown in Table 4) with a PGS variant with a graph kernel logistic regression (GKLR) model as the classifier. The kernels used here are the PyramidMatch (PM) kernel, the shortest-path (SP) kernel, and the Weisfeiler-Lehman (WL) kernel.

| **Dataset** | **Model** | | | | **Subscores** | |
|---|---|---|---|---|---|---|
| | | **PGS (↓)** | **PGS-GKLR (↓)** | **PM (↓)** | **SP (↓)** | **WL (↓)** |
| Planar-L | AutoGraph | **34.0** ± 1.8 | **6.2** ± 2.1 | 5.3 ± 1.4 | **5.2** ± 0.9 | **6.7** ± 1.9 |
| | DiGress | 45.2 ± 1.8 | 22.7 ± 0.9 | 19.3 ± 0.5 | 22.8 ± 0.6 | 20.5 ± 0.6 |
| | GRAN | 99.7 ± 0.2 | 43.1 ± 0.3 | 8.8 ± 0.8 | 5.2 ± 2.4 | 43.1 ± 0.3 |
| | ESGG | 45.0 ± 1.4 | 14.4 ± 1.0 | **2.7** ± 2.3 | 12.8 ± 0.7 | 14.6 ± 0.8 |
| Lobster-L | AutoGraph | 18.0 ± 1.6 | 10.6 ± 1.2 | 10.3 ± 0.9 | 8.4 ± 1.4 | 10.5 ± 1.7 |
| | DiGress | **3.2** ± 2.6 | **2.4** ± 2.5 | **2.6** ± 1.7 | **2.5** ± 2.2 | **2.2** ± 2.4 |
| | GRAN | 85.4 ± 0.5 | 72.7 ± 0.8 | 52.3 ± 0.8 | 57.9 ± 1.2 | 72.7 ± 0.8 |
| | ESGG | 69.9 ± 0.6 | 56.1 ± 0.6 | 42.0 ± 0.6 | 41.8 ± 1.0 | 56.1 ± 0.6 |
| SBM-L | AutoGraph | **5.6** ± 1.5 | **5.7** ± 1.1 | **1.4** ± 1.5 | 5.7 ± 1.1 | **1.3** ± 2.0 |
| | DiGress | 17.4 ± 2.3 | 8.8 ± 2.4 | 7.8 ± 2.4 | **4.0** ± 2.2 | 9.0 ± 2.5 |
| | GRAN | 69.1 ± 1.4 | 47.4 ± 1.0 | 46.8 ± 1.0 | 32.7 ± 1.3 | 47.4 ± 1.0 |
| | ESGG | 99.4 ± 0.2 | 93.5 ± 0.3 | 23.8 ± 1.8 | 93.5 ± 0.3 | 42.6 ± 1.1 |
| Proteins | AutoGraph | **67.7** ± 7.4 | 39.2 ± 2.8 | 14.0 ± 2.5 | 39.2 ± 2.8 | 16.5 ± 2.1 |
| | DiGress | 88.1 ± 3.1 | 44.8 ± 1.3 | **3.6** ± 3.0 | 44.8 ± 1.3 | **8.9** ± 3.3 |
| | GRAN | 89.7 ± 2.7 | 59.4 ± 2.0 | 55.0 ± 1.8 | 45.7 ± 1.9 | 59.4 ± 2.0 |
| | ESGG | 79.2 ± 4.3 | **31.9** ± 5.0 | 17.7 ± 2.3 | **31.9** ± 5.0 | 22.0 ± 2.1 |