A Extended Dataset Description and Results

A.1 Optional variants of `PascalVOC-SP` and `COCO-SP` datasets

For `PascalVOC-SP` and `COCO-SP` datasets, the graphs that we keep by default are the `rag-boundary` graphs which are based on SLIC superpixels extraction with compactness value of 30. Additionally, we provide optional variants of both these datasets which are based on SLIC superpixels extraction with compactness value of 10, and two other graph construction formats, `coo` and `coo-feat`. In this section, we include the description and results of baseline experiments of these optional datasets as well. Note that any of these version of the `SP` dataset can be used as independent LRGB dataset.

Construction of `coo` and `coo-feat` graphs: Under these two construction methods, the resultant graphs are 8 nearest neighbor graphs where the pairwise adjacency weights for two nodes are *first* constructed based on coordinates (for `coo`) or based on coordinates and feature intensities (for `coo-feat`) of the superpixels nodes, and *then* each node is directly connected to 8 other nodes with the highest weight scores. The weights computation is based on the Eqn. 1 for `coo` and Eqn. 2 for `coo-feat`:

$$W_{ij}^{8-nn} = \exp\left(-\frac{\|x_i - x_j\|}{\sigma_x^2}\right) \tag{1}$$

$$W_{ij}^{8-nn} = \exp\left(-\frac{\|x_i - x_j\|}{\sigma_x^2} - \frac{\|f_i - f_j\|}{\sigma_f^2}\right) \tag{2}$$

where, x_i, x_j are the 2 dimensional coordinates, and f_i, f_j are the 12 dimensional (3 dimensions each of mean, std, max, min) RGB feature values of superpixels i, j respectively, σ_x^2 is a scaling parameter defined as the average distance x_k of the $k = 8$ nearest neighbors for each node. The initial feature of each superpixel node is 12 dimensional RGB feature value (mean, std, max, min) and that of an edge between two nodes is a 1 dimensional edge weight that is given by Eqn. 1 or 2 for the respective graph format.

Statistics and Baseline Results. The complete statistics of the aforementioned optional versions of the `PascalVOC-SP` and `COCO-SP` datasets are included in Table A.1. Note that that versions with the options **SLIC**: 30 and **Graph**: `rag-boundary` is the default dataset for both `PascalVOC-SP` and `COCO-SP` that we present in the main Section 3. The results of the baseline experiments on all the dataset versions are reported in Table A.2.

Table A.1: Statistics of 6 tried versions of `PascalVOC-SP` and `COCO-SP` datasets, each derived with a different combination of **Options** in terms of: (i) **SLIC**, which denotes the value of `compactness` parameter used during the extraction of superpixels by SLIC algorithm [1] and (ii) **Graph**, which denotes the graph format that was used to construct the adjacency matrix. The **Graph** options '`coo`' refers to a 8-nn graph where the edge weight is based on superpixel coordinates (Eqn. 1), '`coo-feat`' refers to a 8-nn graph where the edge weight is based on superpixel coordinates as well as feature intensities (Eqn. 2), '`rag-boundary`' refers to a region boundary graph.

Dataset	Options SLIC	Options Graph	Total Graphs	Total Nodes	Avg Nodes	Mean Deg.	Total Edges	Avg Edges	Avg Short.Path.	Avg Diameter
`PascalVOC-SP`	10	`coo`	11,355	4,747,374	418.09	8.00	37,978,992	3,344.69	7.50±0.76	17.89±2.10
		`coo-feat`				8.00	37,978,992	3,344.69	7.50±0.76	17.89±2.10
		`reg-bound`				5.62	26,659,158	2,347.79	9.08±1.23	22.99±3.70
	30	`coo`	11,355	5,443,545	479.40	8.00	43,548,360	3,835.17	8.05±0.18	19.40±0.65
		`coo-feat`				8.00	43,548,360	3,835.17	8.05±0.18	19.40±0.65
		`reg-bound`				5.65	30,777,444	2,710.48	10.74±0.51	27.62±2.13
`COCO-SP`	10	`coo`	123,286	49,732,322	403.39	8.00	397,858,524	3,227.12	7.39±0.77	17.61±2.12
		`coo-feat`				8.00	397,858,524	3,227.12	7.39±0.77	17.61±2.12
		`reg-bound`				5.61	278,816,918	2,261.55	8.85±1.23	22.40±3.70
	30	`coo`	123,286	58,793,216	476.88	8.00	470,345,728	3,815.08	8.06±0.18	19.42±0.70
		`coo-feat`				8.00	470,345,728	3,815.08	8.06±0.18	19.42±0.70
		`reg-bound`				5.65	332,091,902	2,693.67	10.66±0.55	27.39±2.14

Table A.2: Baseline experiments for `PascalVOC-SP` and `COCO-SP` for node classification task. Performance metric is macro F1 on the respective splits (Higher is better). All experiments are run 4 times with 4 different seeds. All models have approximately 500k learnable parameters for fair comparison. The MP-GNN models are 8 layers deep, while the transformer-based models have 4 layers in order to maintain comparable hidden representation size at the fixed parameter budget.

Dataset (SLIC)		Model	# Params	coo		coo-feat		reg-bound	
				Train F1	Test F1 ↑	Train F1	Test F1 ↑	Train F1	Test F1 ↑
PascalVOC-SP	10	GCN	496k	0.1559±0.0079	0.1281±0.0025	0.1956±0.0202	0.1321±0.0043	0.1530±0.0048	0.1306±0.0025
		GINE	505k	0.2178±0.0382	0.1127±0.0039	0.3007±0.0461	0.1078±0.0035	0.2278±0.0224	0.1231±0.0052
		GatedGCN	502k	0.4319±0.0187	0.2788±0.0032	0.3560±0.0567	0.2289±0.0137	0.3574±0.0573	0.2705±0.0251
		GatedGCN+LapPE	502k	0.4390±0.0144	0.2803±0.0031	0.3535±0.0376	0.2241±0.0035	0.3553±0.0396	0.2722±0.0149
		Transformer+LapPE	501k	0.6140±0.0635	0.2661±0.0129	0.5594±0.0445	0.2667±0.0060	0.5925±0.0447	0.2627±0.0086
		SAN+LapPE	531k	0.6691±0.0339	0.2904±0.0031	0.5555±0.0650	0.2808±0.0047	0.5636±0.0506	0.3031±0.0046
		SAN+RWSE	468k	0.6200±0.0502	0.2841±0.0090	0.5726±0.0615	0.2764±0.0184	0.5968±0.0487	0.3113±0.0072
	30	GCN	496k	0.1469±0.0068	0.1262±0.0031	0.1742±0.0042	0.1326±0.0015	0.1450±0.0125	0.1268±0.0060
		GINE	505k	0.2575±0.0283	0.1203±0.0045	0.2479±0.0318	0.1035±0.0015	0.2088±0.0268	0.1265±0.0076
		GatedGCN	502k	0.4311±0.0325	0.2916±0.0058	0.3379±0.0107	0.2410±0.0015	0.3552±0.0451	0.2873±0.0219
		GatedGCN+LapPE	502k	0.4223±0.0356	0.2890±0.0057	0.3110±0.0706	0.2317±0.0217	0.3512±0.0167	0.2860±0.0085
		Transformer+LapPE	501k	0.6213±0.0393	0.2633±0.0056	0.6607±0.0684	0.2697±0.0081	0.7170±0.0048	0.2694±0.0098
		SAN+LapPE	531k	0.6485±0.0711	0.3218±0.0160	0.5242±0.0480	0.3003±0.0046	0.5723±0.0427	0.3230±0.0039
		SAN+RWSE	468k	0.6240±0.0866	0.3227±0.0084	0.5869±0.0349	0.3124±0.0091	0.5819±0.0331	0.3216±0.0027
COCO-SP	10	GCN	509k	0.0852±0.0030	0.0770±0.0017	0.0919±0.0058	0.0780±0.0026	0.0885±0.0078	0.0809±0.0043
		GINE	515k	0.1874±0.0071	0.1109±0.0048	0.1605±0.0090	0.0846±0.0045	0.1812±0.0155	0.1196±0.0053
		GatedGCN	508k	0.3009±0.0078	0.2280±0.0032	0.2842±0.0077	0.2130±0.0036	0.3149±0.0099	0.2542±0.0044
		GatedGCN+LapPE	509k	0.3018±0.0057	0.2307±0.0014	0.2789±0.0080	0.2110±0.0036	0.3184±0.0144	0.2529±0.0063
		Transformer+LapPE	508k	0.3700±0.0141	0.2455±0.0036	0.3775±0.0082	0.2492±0.0036	0.3758±0.0205	0.2478±0.0068
		SAN+LapPE	536k	0.3437±0.0096	0.2605±0.0062	0.3278±0.0041	0.2596±0.0015	0.2541±0.0394	0.2325±0.0191
		SAN+RWSE	474k	0.3557±0.0264	0.2675±0.0126	0.3270±0.0145	0.2585±0.0046	0.2815±0.0371	0.2442±0.0231
	30	GCN	509k	0.0914±0.0056	0.0797±0.0026	0.1003±0.0043	0.0843±0.0019	0.0948±0.0014	0.0841±0.0010
		GINE	515k	0.1742±0.0186	0.1168±0.0053	0.1646±0.0081	0.1003±0.0022	0.2100±0.0041	0.1339±0.0044
		GatedGCN	508k	0.3024±0.0043	0.2441±0.0035	0.2926±0.0154	0.2285±0.0069	0.3167±0.0059	0.2641±0.0045
		GatedGCN+LapPE	509k	0.3101±0.0062	0.2454±0.0015	0.2894±0.0060	0.2283±0.0036	0.3102±0.0112	0.2574±0.0034
		Transformer+LapPE	508k	0.3855±0.0185	0.2579±0.0057	0.3750±0.0224	0.2589±0.0069	0.3912±0.0098	0.2618±0.0031
		SAN+LapPE	536k	0.3376±0.0455	0.2781±0.0143	0.2941±0.0810	0.2498±0.0513	0.2830±0.0246	0.2592±0.0158
		SAN+RWSE	474k	0.3652±0.0104	0.2817±0.0047	0.3754±0.0074	0.2869±0.0067	0.2657±0.0224	0.2434±0.0156

A.2 Extended description of `Peptides-struct`

Below, we describe the 11 tasks from the `Peptides-struct` dataset, which represented properties computed from the 3D structure, then normalized to zero mean and unit standard deviation.

- **Inertia_mass** The inertia of the molecules according to its 3 principal components, using the mass of the atoms and their distances to the centroid.
- **Inertia_valence** The inertia of the molecules according to its 3 principal components, using the valence of the atoms and their distances to the centroid.
- **Length** The maximum distance between each atom-pairs in each of its 3 main axes.
- **Sphericity** A measure of how much the molecule looks like a sphere: the ratio of the molecule's surface area to the surface area of a sphere with similar volume.
- **Plane_best_fit** The average distance of all heavy atoms from the plane of best fit.

A.3 Extended Results for `Peptides-struct`

Table A.3: Extended evaluation metrics for `Peptides-struct`. The training and testing performance is quantified in terms of the Coefficient of Determination R^2, in addition to MAE reported in Table 4.

Model	# Params.	Train MAE	Test MAE ↓	Train R2	Test R2 ↑
GCN	508k	0.2939±0.0055	0.3496±0.0013	0.6513±0.0078	0.6019±0.0027
GCNII	505k	0.2957±0.0025	0.3471±0.0010	0.6913±0.0242	0.6148±0.0054
GINE	476k	0.3116±0.0047	0.3547±0.0045	0.6494±0.0108	0.5943±0.0067
GatedGCN	509k	0.2761±0.0032	0.3420±0.0013	0.6907±0.0058	0.6254±0.0013
GatedGCN+RWSE	506k	0.2578±0.0116	0.3357±0.0006	0.7204±0.0149	0.6329±0.0034
Transformer+LapPE	488k	0.2403±0.0066	**0.2529±0.0016**	0.8027±0.0108	**0.7743±0.0053**
SAN+LapPE	493k	0.2822±0.0108	0.2683±0.0043	0.6887±0.0153	0.7581±0.0057
SAN+RWSE	500k	0.2680±0.0038	0.2545±0.0012	0.7112±0.0049	0.7716±0.0034

A.4 Correlation of labels in `Peptides-func` and `Peptides-struct`

The `Peptides-func` is a multi-label classification and `Peptides-struct` is a multi-label regression. Evaluating the correlation between the labels will ensure that labels are not redundant and provide a variety of information, Figure A.1. We observe that there is very little correlation between classes of peptides. For the structural dataset, there are some correlations, which are expected since intertia is related to length and spherocity, but in general, the correlation remains limited, which motivates using a multi-label regression.

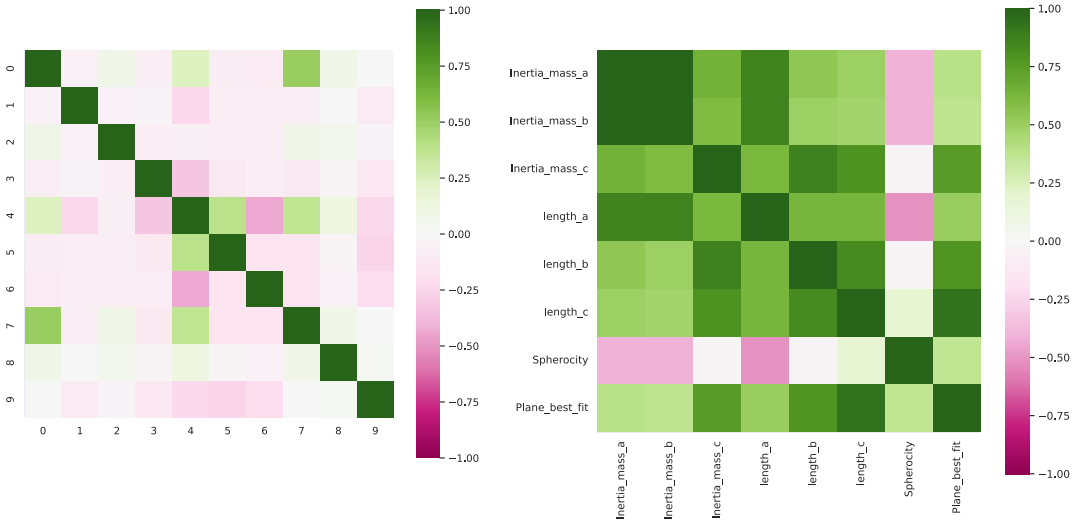

Figure A.1: **Left:** Visualization of pearson correlation between classes in peptides-func dataset. **Right:** Visualization of correlation between geometric properties in peptides-struct dataset.

A.5 Dataset Licenses.

The information on the dataset sources that we used for the proposed LRGB datasets' preparation, the original licenses of use and the release licenses are in Table A.4.

Table A.4: Original resources that our 5 datasets are derived from and their licensing information. *Custom License for Pascal VOC 2011 (respecting Flickr terms of use).

	Derived from	Original License	Release License
`PascalVOC-SP`	Pascal VOC [18]	Custom*	Custom*
`COCO-SP`	MS COCO [38]	CC BY 4.0	CC BY 4.0
`PCQM-Contact`	PCQM4Mv2 [26]	CC BY 4.0	CC BY 4.0
`Peptides-func`	SATPdb [54]	CC BY-NC 4.0	CC BY-NC 4.0
`Peptides-struct`	SATPdb [54]	CC BY-NC 4.0	CC BY-NC 4.0

B Visualizations

B.1 PascalVOC-SP

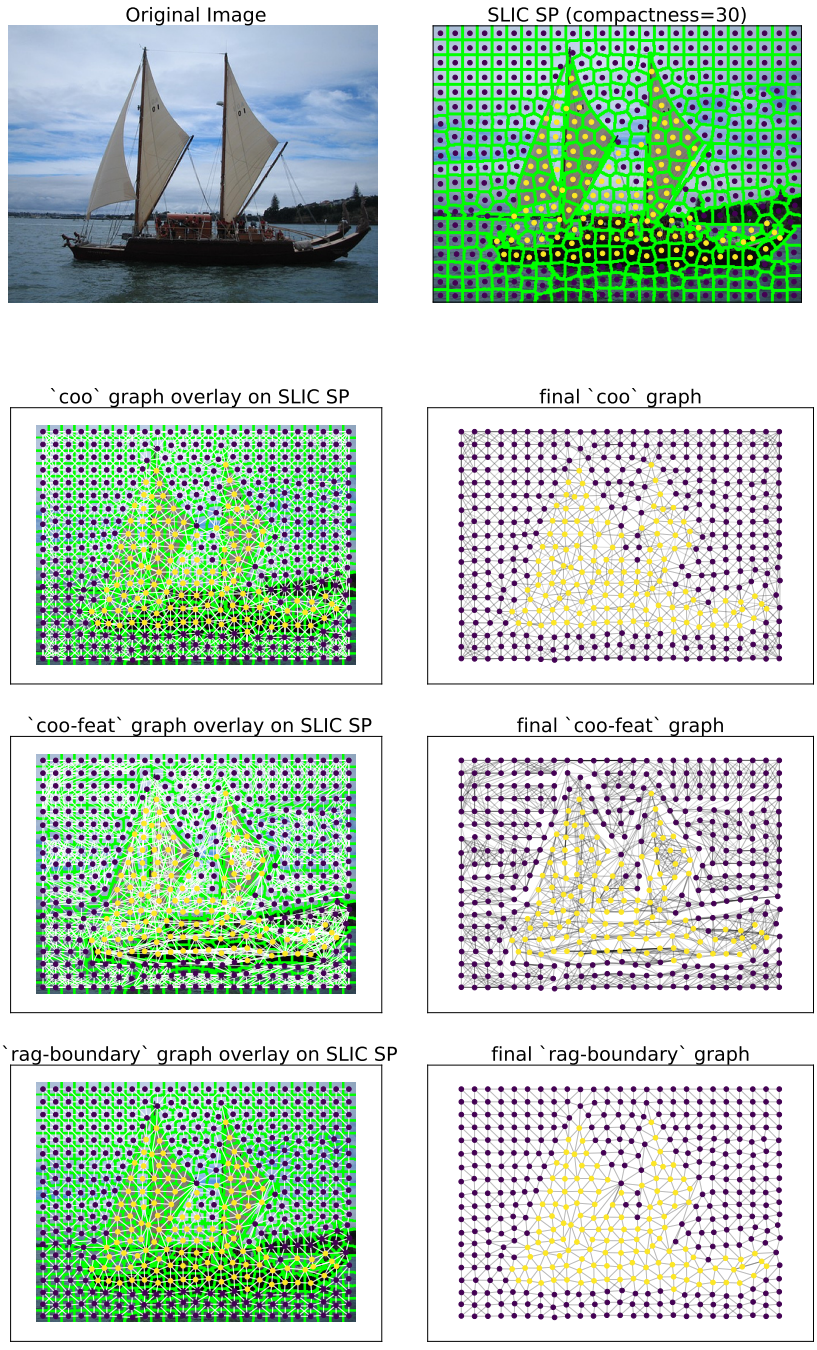

Figure B.1: Visualizations of a sample image and its SP graphs from PascalVOC-SP dataset with 465 nodes and 3,720 edges each for coo, coo-feat graph and 2,628 edges for reg-bound graph. Unique colors on the nodes denote the corresponding node labels.

B.2 COCO-SP

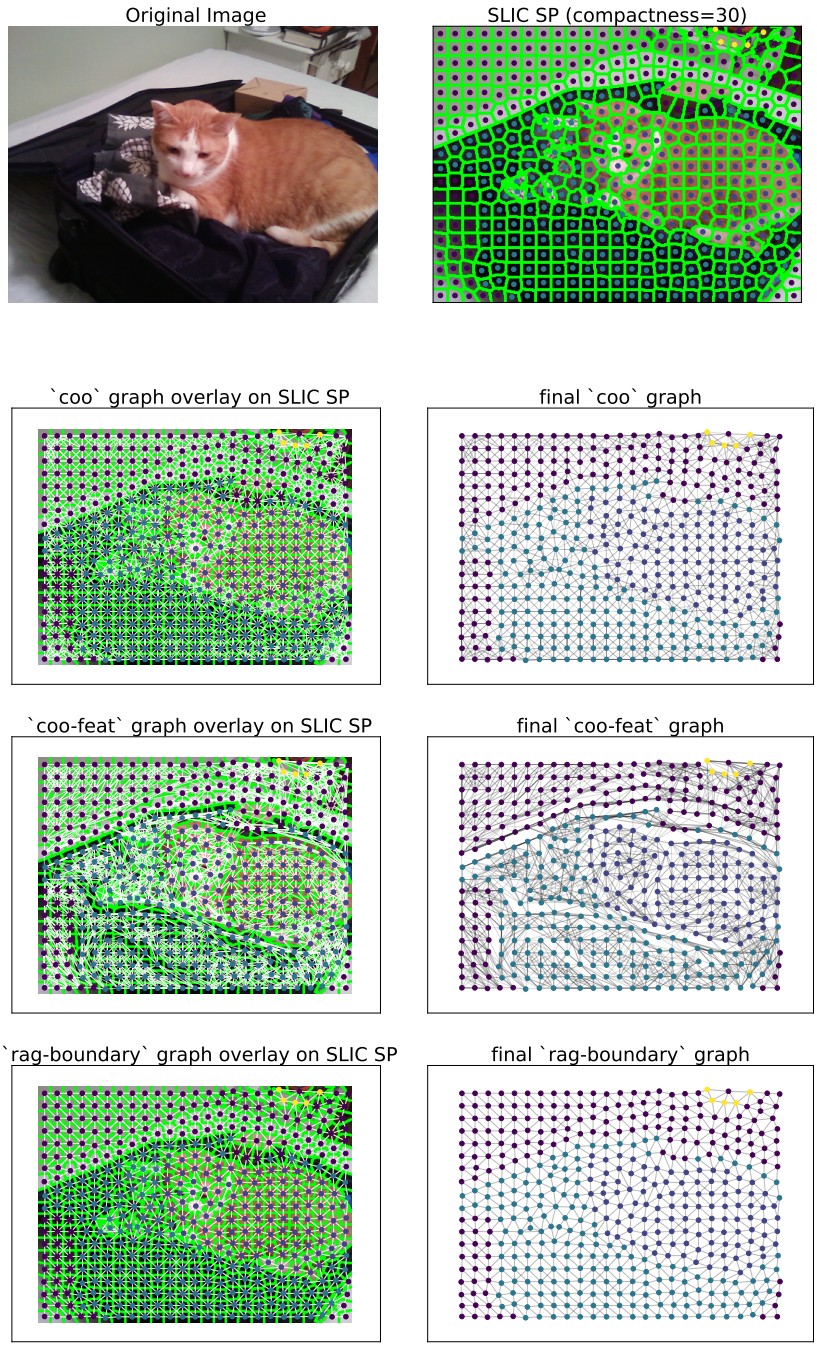

Figure B.2: Visualizations of a sample image and its SP graphs from COCO-SP dataset with 470 nodes and 3,760 edges each for coo, coo-feat graph and 2,662 edges for reg-bound graph. Unique colors on the nodes denote the corresponding node labels.

B.3 `Peptides-func` and `Peptides-struct`

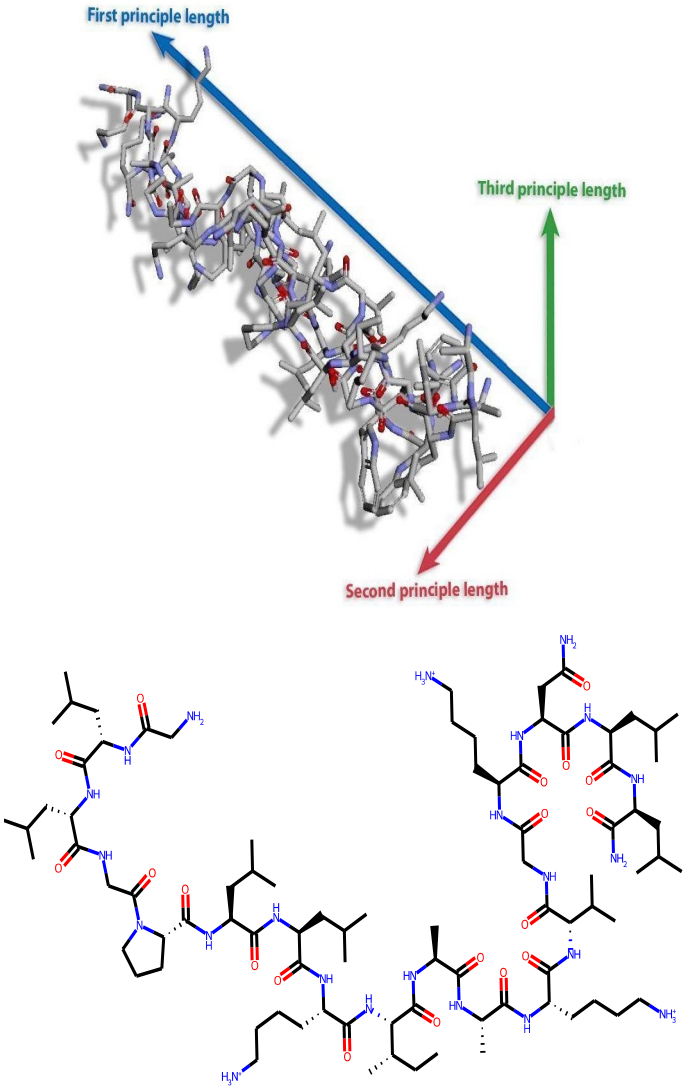

Figure B.3: Large size visualization of Figure 3. **Top:** 3D Visualization of "GLLGPLLKI-AAKVGKNLL" peptide. **Bottom:** The molecular graph for the same peptide.

C Experimental Details

Table C.1: Baseline hyperparameters of 7 evaluated models on the 5 new LRGB benchmarks. Shown is the size of the hidden node representation d and the number of layers L. Where applicable, the type of positional/structural embedding is shown: LapPE-k denotes Laplacian positional encoding [34] with first k non-trivial eigenvectors (with original Transformer-based encoder for SAN [34] and more parameter-efficient DeepSet encoder for GatedGCN); RWSE-m denotes random-walk structural encoding [15] with $1..m$ steps and a linear encoder.

	`PascalVOC-SP`	`COCO-SP`	`PCQM-Contact`	`Peptides-func`	`Peptides-struct`
GCN	d=220, L=8	d=220, L=8	d=275, L=5	d=300, L=5	d=300, L=5
GCNII	d=220, L=8	d=220, L=8	d=275, L=5	d=300, L=5	d=300, L=5
GINE	d=166, L=8	d=166, L=8	d=208, L=5	d=208, L=5	d=208, L=5
GatedGCN(+PE/SE)	d=108, L=8	d=108, L=8	d=138, L=5	d=138, L=5	d=138, L=5
used PE/SE	LapPE-10	LapPE-10	RWSE-16	RWSE-16	RWSE-16
Transformer+LapPE	d=120, L=4	d=120, L=4	d=120, L=4	d=120, L=4	d=120, L=4
	LapPE-10	LapPE-10	LapPE-10	LapPE-10	LapPE-10
SAN+LapPE	d=88, L=4	d=88, L=4	d=84, L=4	d=84, L=4	d=84, L=4
	LapPE-10	LapPE-10	LapPE-10	LapPE-10	LapPE-10
SAN+RWSE	d=96, L=4	d=96, L=4	d=100, L=4	d=100, L=4	d=100, L=4
	RWSE-16	RWSE-16	RWSE-16	RWSE-16	RWSE-16

C.1 Details on Baseline Experiments Setup

Models. We use GCN [33], GCNII [11], GINE [62, 28] and GatedGCN [8] models from the local MP-GNNs class, and fully connected Transformer [58] with Laplacian PE (LapPE) [14, 13] and SAN [34] models among the Transformer class. The GCN (Graph Convolutional Network) is the most popularly used local MP-GNN baseline, GCNII [11] is an extension of the vanilla GCN, GINE (Graph Isomorphism Network) is a 1-WL expressive MP-GNN with ability to incorporate edge features into its update equation [28], and GatedGCN (Gated Graph Convolutional Network) is a soft-attention based GCN which uses learned edge gates to improve the aggregation procedure. Transformer with LapPE is a generalization of the vanilla Transformer network [58] from Natural Language Processing (NLP) domain to graphs and SAN (Spectral Attention Network) is a powerful fully-connected Graph Transformer which includes a learned PE module based on Laplacian eigenvectors and eigenvalues, alongside separate treatment of real and non-real graph edges [34]. We use SAN with LapPE as well RWSE (Random Walk Structural Encoding) [15]. The collection of above baseline models allows us to show performance trends using simple, straightforward models such as GCN and Transformer to advanced ones such as GatedGCN and SAN. We believe this baseline collection, albeit small, represents a diverse representation of the course of action graph deep learning has evolved to, reaching at a stage where we can embark conveniently towards the development of GNNs that learn efficiently to propagate long-range dependencies.

Experimental Setup. In order to facilitate fair comparison and reliable discussion of the observed trends, we select the hyperparameters of the aforementioned baselines such that they yield models within a budget of approx. 500k learnable parameters. To this end, we configure 4-8 layers deep models and adjust their hidden dimension size accordingly to the 500k parameter budget. For a list of hyperparameters used in each baseline see Table C.1. We run each experiment 4 times with different random seeds and report the mean and standard deviation of the respective performance metrics.

For optimization, we use Adam [32] with default settings. We set the starting learning rate between 0.0003 and 0.001 depending on the model and dataset, and decay it by 0.5 factor upon reaching a validation loss plateau. We limit the training time up to 60h, which is adequate for the models to converge, except SAN on `COCO-SP`. SAN is particularly computationally intensive and may require a week of single NVidia A100 computation time to converge on `COCO-SP`. Individual configuration files with exact hyperparameters for all 7 models and 5 datasets are provided with the source code.

C.2 Computing environment and used resources

Our implementation uses GraphGPS [51] built on PyG and its GraphGym module [19, 67] that are all released under MIT License. All presented experiments were executed in a shared computing cluster environment (Digital Research Alliance of Canada and Mila Quebec AI Institute) with multiple CPU and GPU architectures: NVidia V100 (32GB), NVidia RTX8000 (48GB), and NVidia A100 (40GB). The resource budget for each experiment was 1 GPU, 4 CPUs, and up to 32GB system RAM. Except `COCO-SP`, which required up to 72GB RAM. Average run times are shown in Table C.2.

Table C.2: Wall-clock run times. Average epoch time (average of 5 epochs, including validation performance evaluation) is shown for each model and dataset combination. Additionally, the precomputation time needed for LapPE and RWSE statistics is listed in the bottom of the table. The times were measured on a single NVidia A100 GPU system with 4 CPU cores of AMD Milan 7413.

avg. time / epoch	`PascalVOC-SP`	`COCO-SP`	`PCQM-Contact`	`Peptides-func`	`Peptides-struct`
GCN	8.8s	111s	138s	3.0s	2.6s
GCNII	8.2s	106s	137s	2.7s	2.4s
GINE	7.2s	91s	138s	2.5s	2.6s
GatedGCN	12s	151s	138s	3.3s	3.3s
Transformer+LapPE	13s	154s	145s	5.8s	5.9s
SAN+LapPE	179s	2190s	793s	54.7s	53.6s
SAN+RWSE	165s	2014s	740s	49.1s	49.7s
LapPE precomp.	8min 40s	1h 34min	5min 18s	1min 13s	1min 14s
RWSE precomp.	7min 51s	1h 24min	6min 29s	53s	53s

D Additional Experiments with $L = 2$ MP-GNNs

Here we investigate a shallow but wider variation of the baseline MP-GNN models. Instead of 5 or 8 layers (Table C.1) we consider 2-layer MP-GNN architectures that allow for larger hidden node representations within the 500k parameter budget, Table D.1. These provide additional set of baselines, that investigate whether this limited receptive field of MP-GNNs is to a detriment in the proposed LRGB datasets, and also, whether the deeper architectures, evaluated in the main text, are not suffering from catastrophic over-smoothing and/or over-squashing. Note, that we do use residual connections in all our baseline models, this has been shown to significantly help to prevent deterioration of the models' performance with increasing depth.

Generally, we observe majorily decreased performance of 2-layer MP-GNNs as compared to their deeper versions, while their relative ordering by their performance remains largely the same. This finding confirms that the access to only a narrow receptive field is severely limiting. Additionally, we observe *much increased positive impact* of augmenting 2-layer GatedGCN with positional or structural encodings. GatedGCN with LapPE or RWSE outperforms standard GatedGCN (and any other tested MP-GNN) by a large margin particularly in `PCQM-Contact`, `Peptides-func`, and `Peptides-struct`. In the case of the deeper MP-GNN configurations (Table C.1) this effect is not observed, suggesting that the positional or structural encodings provide additional information beyond the 2-hop neighborhood that a deeper GatedGCN appears to be able to substitute.

Table D.1: Hyperparameters of evaluated *shallow* MP-GNN baseline models. The number of layers is set to L=2 and the size of the hidden node representation d is set to fill 500k parameter budget. Where applicable, the type of positional/structural embedding is shown.

	`PascalVOC-SP`	`COCO-SP`	`PCQM-Contact`	`Peptides-func`	`Peptides-struct`
GCN	d=350, L=2	d=345, L=2	d=380, L=2	d=460, L=2	d=460, L=2
GCNII	d=350, L=2	d=345, L=2	d=380, L=2	d=460, L=2	d=460, L=2
GINE	d=285, L=2	d=285, L=2	d=300, L=2	d=330, L=2	d=330, L=2
GatedGCN(+PE/SE)	d=200, L=2	d=200, L=2	d=210, L=2	d=215, L=2	d=215, L=2
used PE/SE	LapPE-10	LapPE-10	RWSE-16	RWSE-16	RWSE-16

Table D.2: Baseline experiments for `PascalVOC-SP` and `COCO-SP` with `rag-boundary` graph on SLIC compactness 30 for node classification task for MP-GNNs with 2 layers and 500k parameters. Performance metric is macro F1 on the respective splits (Higher is better). All experiments are run 4 times with 4 different seeds.

Model ($L=2$)	**# Params.**	`PascalVOC-SP`		**# Params**	`COCO-SP`	
		Train F1	**Test F1** ↑		**Train F1**	**Test F1** ↑
GCN	504k	0.1014±0.0031	0.1011±0.0024	511k	0.0589±0.0019	0.0562±0.0015
GCNII	503k	0.1137±0.0055	0.1067±0.0028	509k	0.0705±0.0017	0.0656±0.0018
GINE	500k	0.1467±0.0116	0.1238±0.0046	517k	0.1110±0.0043	0.0929±0.0027
GatedGCN	491k	0.2382±0.0313	0.2114±0.0157	503k	0.1581±0.0033	0.1476±0.0027
GatedGCN+LapPE	492k	0.2583±0.0458	0.2232±0.0255	504k	0.1668±0.0037	0.1553±0.0026

Table D.3: Baselines for `Peptides-func` (graph classification) and `Peptides-struct` (graph regression) for MP-GNNs with 2 layers and 500k parameters. Performance metric is Average Precision (AP) for classification and MAE for regression. Each experiment was run with 4 different seeds.

Model ($L=2$)	**# Params.**	`Peptides-func`		`Peptides-struct`	
		Train AP	**Test AP** ↑	**Train MAE**	**Test MAE** ↓
GCN	509k	0.4956±0.0079	0.4566±0.0059	0.3836±0.0019	0.3950±0.0017
GCNII	507k	0.5543±0.0077	0.4894±0.0039	0.3809±0.0020	0.3929±0.0020
GINE	501k	0.5916±0.0189	0.5003±0.0042	0.3730±0.0029	0.3879±0.0011
GatedGCN	508k	0.6085±0.0071	0.5073±0.0036	0.3757±0.0015	0.3905±0.0006
GatedGCN+RWSE	505k	0.7946±0.0148	0.5812±0.0053	0.3173±0.0084	0.3599±0.0007

Table D.4: Baseline performance on `PCQM-Contact` (link prediction) for MP-GNNs with 2 layers and 500k parameters. Each experiment was repeated with 4 different random seeds.

Model ($L=2$)	**# Params.**	**Test Hits@1** ↑	**Test Hits@3** ↑	**Test Hits@10** ↑	**Test MRR** ↑
GCN	500k	0.0588±0.0007	0.1717±0.0011	0.5664±0.0008	0.1939±0.0003
GCNII	499k	0.0651±0.0035	0.1714±0.0026	0.5399±0.0124	0.1944±0.0020
GINE	507k	0.0596±0.0005	0.1718±0.0009	0.5685±0.0006	0.1949±0.0006
GatedGCN	528k	0.0556±0.0008	0.1707±0.0014	0.5734±0.0027	0.1927±0.0010
GatedGCN+RWSE	525k	0.1068±0.0010	0.3383±0.0012	0.8036±0.0008	0.2937±0.0007

E Inspection of Transformer attention

In this section we investigate how a Transformer with LapPE [13] processes the 5 proposed LRGB datasets and an existing MNIST dataset [14]. In particular, we investigate how strongly a Transformer attends to nodes that are at various k distances away from a node v during updating of its representation h_v^ℓ at layer $\ell \in \{0, \dots, L-1\}$. The goal is to probe whether a model capable of global attention, such as the Transformer with LapPE, in fact attends to nodes farther than the local neighborhood of v, while performing better or comparable to local MP-GNN models.

For each dataset, we used a fully trained graph Transformer model with LapPE (using the same hyperparameters and training pipeline as described in Appendix C) and inspected its attention weights on 128 randomly selected test graphs. For each layer ℓ, we plot the average attention weight aggregated by how far in the graph the node is from the perspective of a "focal" node v that is being processed. That is, we show what attention weight on average a node u that is k-hops away from v (shortest-path distance k) gets. Note, that attention to a node at distance $k=0$ denotes the attention of v to self. The resulting bar plots of attention weights in each of the 5 LRGB datasets are shown in Figures E.1-E.5.

Overall the attention distributions vary across datasets and layers, but generally confirm that Transformer exhibits attention patterns beyond local neighborhoods. In `PascalVOC-SP` and `COCO-SP` the first layer ($\ell=0$) shows higher attention to mid- and long-distance nodes over the close-by nodes; this changes in the second and third layers, where attention to close-by nodes is dominant; and finally the last layer exhibits the most even attention distribution with some bias towards close-by nodes.

In PCQM-Contact dataset, the attention distributions are similar, except the first layer ($\ell = 0$) that is much more uniform yet lightly favoring nodes in the first half of the distance range. Finally, in Peptides-func and Peptides-struct the attention distributions are considerably more consistent across the layers and exhibit mostly linear attention weight decay with the growing shortest-path distance between the nodes.

In addition to the above LRGB datasets, we conducted the same inspection of a Transformer model with LapPE on an existing dataset MNIST [14], that we argue is insufficient for benchmarking a model's ability to capture LRIs. We stick to 100k parameter budget, as per standard for this dataset, using 4 layers with hidden node representation size of 52. A Transformer+LapPE model scores 97.89% test accuracy on a random split, and its attention distribution is shown in Figure E.6. Graphs in MNIST dataset have much smaller graph diameter and except for the first layer ($\ell = 0$), the attention is majorly focused on close neighbors that are up to 4-hops away. While on its own a not sufficient proof, the difference in attention distributions between LRGB datasets and MNIST support the viability of proposed LRGB datasets for testing a model's capability to capture interactions beyond limited local neighborhoods.

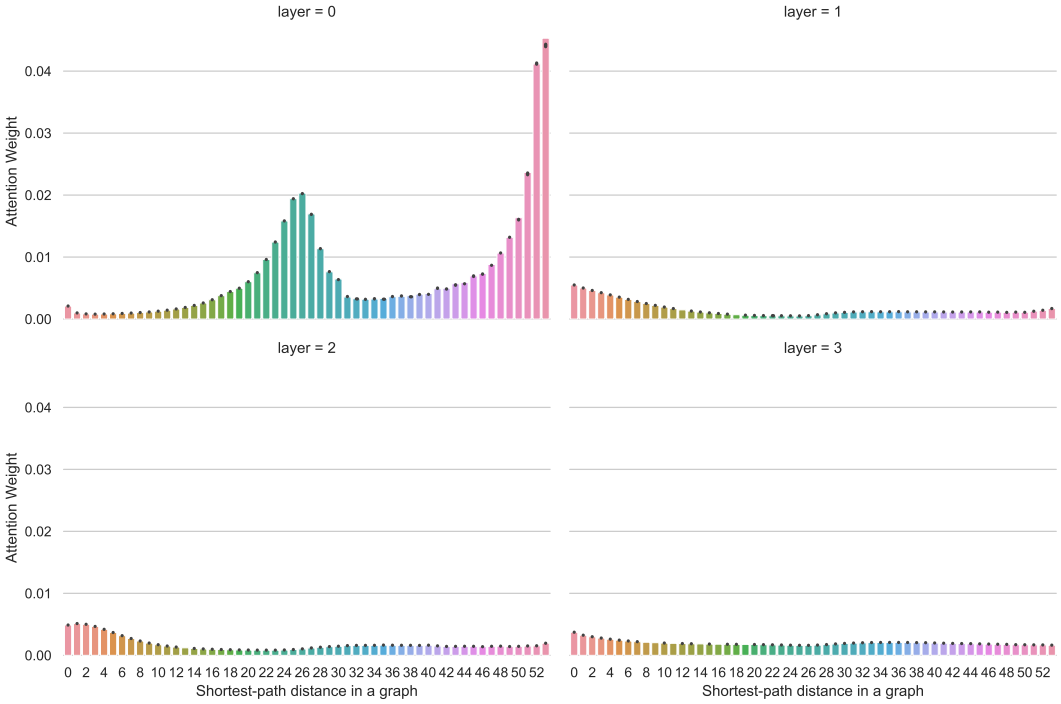

Figure E.1: Average attention weight distribution of Transformer+LapPE on PascalVOC-SP dataset.

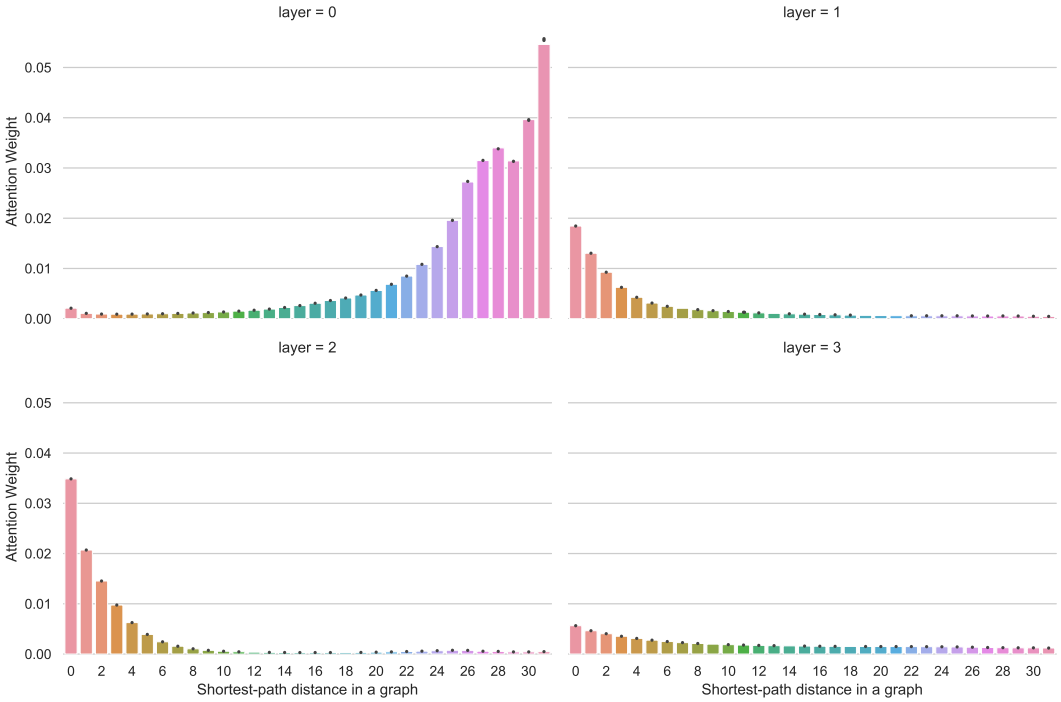

Figure E.2: Average attention weight distribution of Transformer+LapPE on `COCO-SP` dataset.

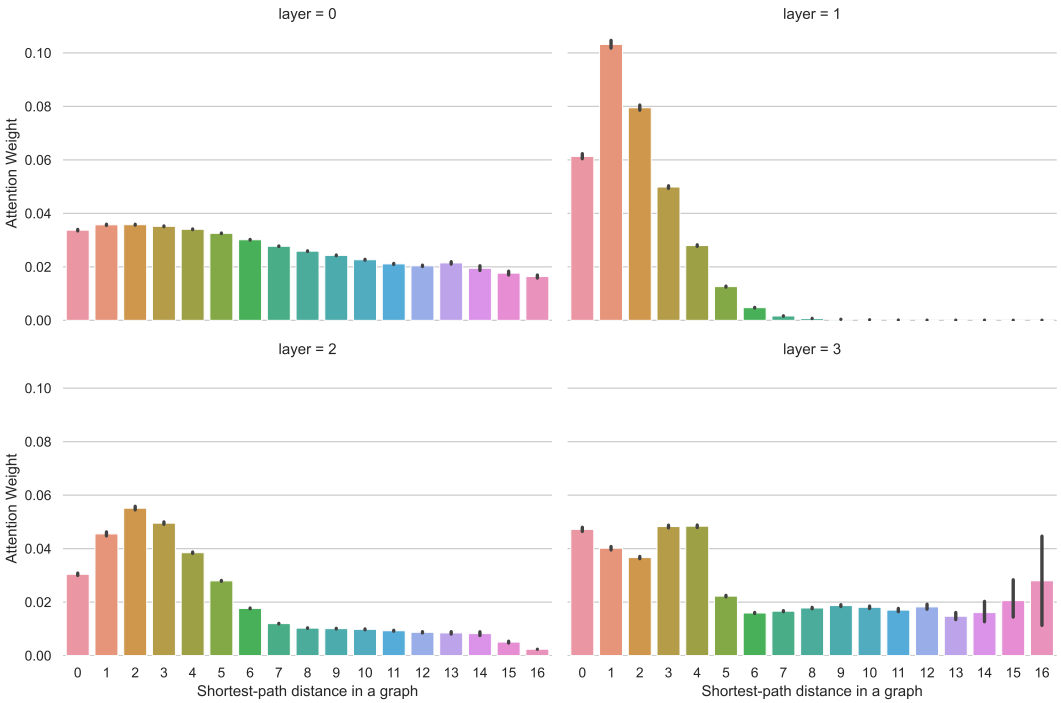

Figure E.3: Average attention weight distribution of Transformer+LapPE on `PCQM-Contact` dataset.

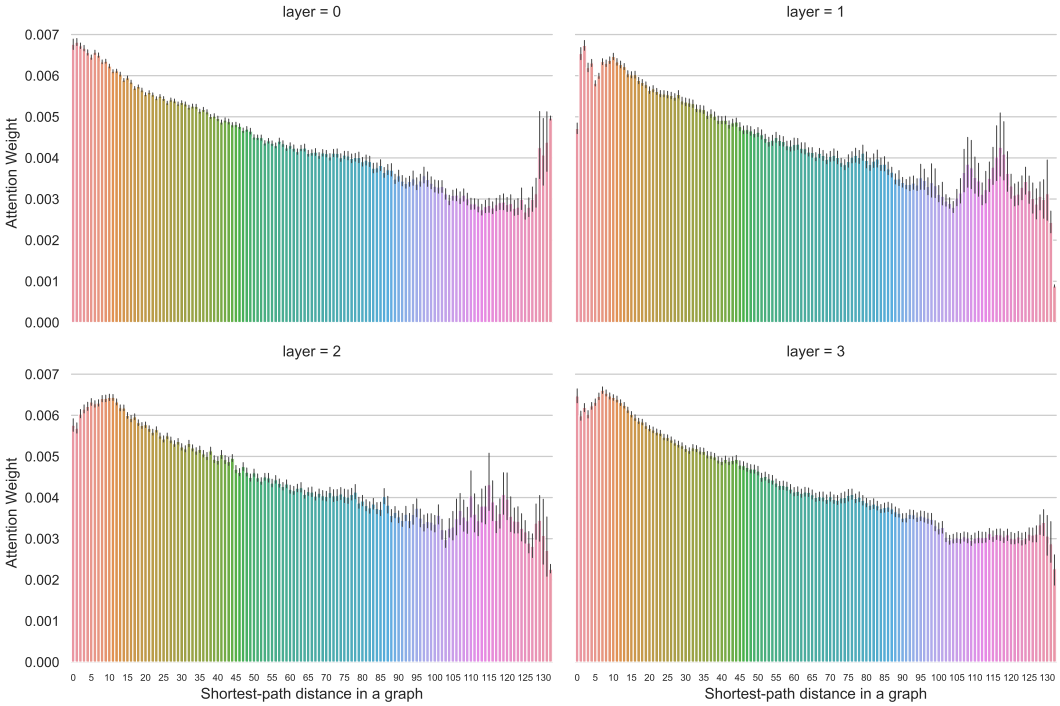

Figure E.4: Average attention weight distribution of Transformer+LapPE on `Peptides-func`.

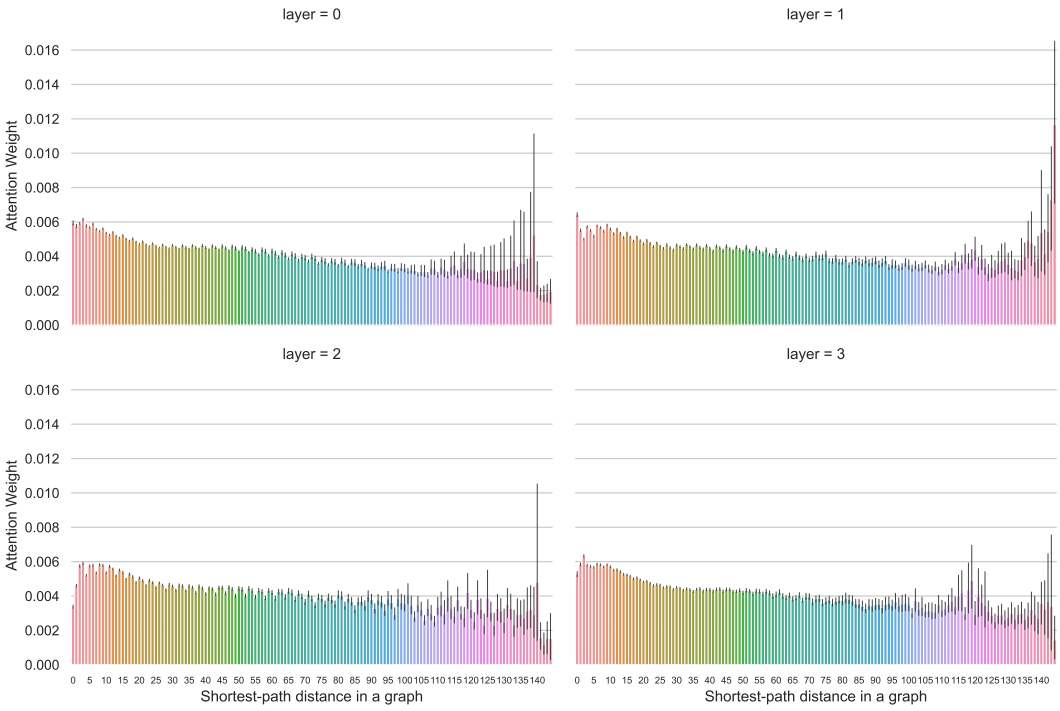

Figure E.5: Average attention weight distribution of Transformer+LapPE on `Peptides-struct`.

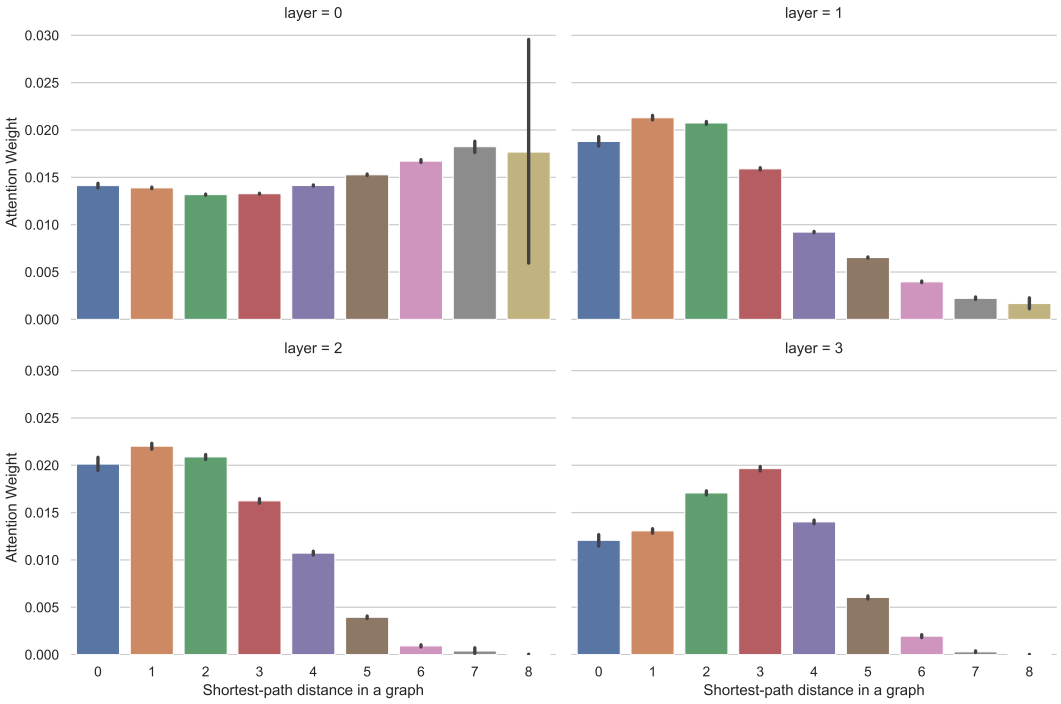

Figure E.6: Average attention weight distribution of Transformer+LapPE on MNIST dataset [14]. Compared to the proposed 5 LRGB datasets, graphs in MNIST dataset have much smaller graph diameter and except the first layer (layer = 0), the attention is majorly focused on close neighbors that are up to 4-hops away.