Dataset	LapPE	L	#Param	Sparse Graph Test Perf.±s.d.	Sparse Graph Train Perf.±s.d.	Sparse Graph #Epoch	Sparse Graph Epoch/Total	Full Graph Test Perf.±s.d.	Full Graph Train Perf.±s.d.	Full Graph #Epoch	Full Graph Epoch/Total
Batch Norm: `False`; Layer Norm: `True`											
ZINC	✗	10	588353	0.278±0.018	0.027±0.004	274.75	26.87s/2.06hr	0.741±0.008	0.431±0.013	196.75	37.64s/2.09hr
	✓	10	588929	0.284±0.012	0.031±0.006	263.00	26.64s/1.98hr	0.735±0.006	0.442±0.031	196.75	31.50s/1.77hr
CLUSTER	✗	10	523146	70.879±0.295	86.174±0.365	128.50	202.68s/7.32hr	19.596±2.071	19.570±2.053	103.00	512.34s/15.15hr
	✓	10	524026	70.649±0.250	86.395±0.528	130.75	200.55s/7.43hr	27.091±3.920	26.916±3.764	139.50	565.13s/22.37hr
PATTERN	✗	10	522742	73.140±13.633	73.070±13.589	184.25	276.66s/13.75hr	50.854±0.111	50.906±0.005	108.00	540.85s/16.77hr
	✓	10	522982	71.005±11.831	71.125±11.977	192.50	294.91s/14.79hr	56.482±3.549	56.565±3.546	124.50	637.55s/22.69hr
Batch Norm: `True`; Layer Norm: `False`											
ZINC	✗	10	588353	0.264±0.008	0.048±0.006	321.50	28.01s/2.52hr	0.724±0.013	0.518±0.013	192.25	50.27s/2.72hr
	✓	10	588929	**0.226±0.014**	0.059±0.011	287.50	27.78s/2.25hr	0.598±0.049	0.339±0.123	273.50	45.26s/3.50hr
CLUSTER	✗	10	523146	72.139±0.405	85.857±0.555	121.75	200.85s/6.88hr	21.092±0.134	21.071±0.037	100.25	595.24s/17.10hr
	✓	10	524026	**73.169±0.622**	86.585±0.905	126.50	201.06s/7.20hr	27.121±8.471	27.192±8.485	133.75	552.06s/20.72hr
PATTERN	✗	10	522742	83.949±0.303	83.864±0.489	236.50	299.54s/19.71hr	50.889±0.069	50.873±0.039	104.50	621.33s/17.53hr
	✓	10	522982	**84.808±0.068**	86.559±0.116	145.25	309.95s/12.67hr	54.941±3.739	54.915±3.769	117.75	683.53s/22.77hr

Table 1: Results of GraphTransformer (GT) on all datasets. Performance Measure for ZINC is MAE, for PATTERN and CLUSTER is Acc. Results (higher is better for all except ZINC) are averaged over 4 runs with 4 different seeds. **Bold**: the best performing model for each dataset. We perform each experiment with given graphs **(Sparse Graph)** and **(Full Graph)** in which we create full connections among all nodes; For ZINC full graphs, edge features are discarded given our motive of the full graph experiments without any sparse structure information.

Model	ZINC	CLUSTER	PATTERN
GNN BASELINE SCORES from (**?**)			
GCN	0.367±0.011	68.498±0.976	71.892±0.334
GAT	0.384±0.007	70.587±0.447	78.271±0.186
GatedGCN	0.214±0.013	76.082±0.196	86.508±0.085
OUR RESULTS			
GT (Ours)	0.226±0.014	73.169±0.622	84.808±0.068

Table 2: Comparison of our best performing scores (from Table 1) on each dataset against the GNN baselines (GCN (**?**), GAT (**?**), GatedGCN(**?**)) of 500k model parameters. **Note:** Only GatedGCN and GT models use the available edge attributes in ZINC.

Dataset	PE	#Param	Sparse Graph Test Perf.±s.d.	Sparse Graph Train Perf.±s.d.	Sparse Graph #Epoch	Sparse Graph Epoch/Total
Batch Norm: `True`; Layer Norm: `False`; L = 10						
ZINC	✗	588353	0.264±0.008	0.048±0.006	321.50	28.01s/2.52hr
	L	588929	**0.226±0.014**	0.059±0.011	287.50	27.78s/2.25hr
	W	590721	0.267±0.012	0.059±0.010	263.25	27.04s/2.00hr
CLUSTER	✗	523146	72.139±0.405	85.857±0.555	121.75	200.85s/6.88hr
	L	524026	**73.169±0.622**	86.585±0.905	126.50	201.06s/7.20hr
	W	531146	70.790±0.537	86.829±0.745	119.00	196.41s/6.69hr
PATTERN	✗	522742	83.949±0.303	83.864±0.489	236.50	299.54s/19.71hr
	L	522982	**84.808±0.068**	86.559±0.116	145.25	309.95s/12.67hr
	W	530742	75.489±0.216	97.028±0.104	109.25	310.11s/9.73hr

Table 3: Analysis of GraphTransformer (GT) using different PE schemes. Notations **x**: No PE; **L**: LapPE (ours); **W**: WL-PE (**?**). **Bold**: the best performing model for each dataset.

they outperform the WL-PE. Besides, WL-PEs tend to overfit SBM datasets and lead to poor generalization.

5 Conclusion

This work presented a simple yet effective approach to generalize transformer networks on arbitrary graphs and introduced the corresponding architecture. Our experiments consistently showed that the presence of – i) Laplacian eigenvectors as node positional encodings and – ii) batch normalization, in place of layer normalization, around the transformer feed forward layers enhanced the transformer universally on all experiments. Given the simple and generic nature of our architecture and competitive performance against standard GNNs, we believe the proposed model can be used as baseline for further improvement across graph applications employing node attention. In future works, we are interested in building upon the graph transformer along aspects such as efficient training on single large graphs, applicability on heterogeneous domains, etc., and perform efficient graph representation learning keeping in account the recent innovations in graph inductive biases.

Acknowledgments

XB is supported by NRF Fellowship NRFF2017-10.

Ad quidem enim iste dolorem, assumenda a necessitatibus quasi architecto, pariatur atque enim accusantium incidunt aliquam est blanditiis ut ullam labore minima?Recusandae blanditiis asperiores tenetur nobis distinctio laborum architecto, doloremque voluptates nesciunt optio officia laborum sed cupiditate labore numquam ipsum dolorem, aut atque deleniti ullam voluptates recusandae voluptatibus tempora numquam tenetur sunt, excepturi vero dicta placeat ad aspernatur assumenda voluptas?Aliquam totam officia, reprehenderit non unde quam dignissimos, quo alias maxime asperiores nemo deserunt?Mollitia ex nostrum, accusamus natus doloremque veniam nesciunt laboriosam ullam nemo illum illo, aliquid animi natus dolorum facilis perspiciatis est quo illum?Quia molestias sit dolorem possimus, unde mollitia rem culpa, veritatis impedit quia eum minima neque officia veniam id consequatur laudantium, corporis harum quas consequatur eos asperiores aliquam natus deserunt ad aliquid?Fuga laborum sint rerum ipsam aliquam modi, blanditiis iste quis ducimus, itaque consequatur eveniet.Dignissimos voluptates omnis pariatur nam harum magnam animi, eveniet ut nihil architecto qui possimus cupiditate hic velit.Maiores saepe animi maxime similique laudantium, ipsa ea sint esse est necessitatibus