Graph Your Own Prompt

Xi Ding, Lei Wang, Piotr Koniusz, Yongsheng Gao

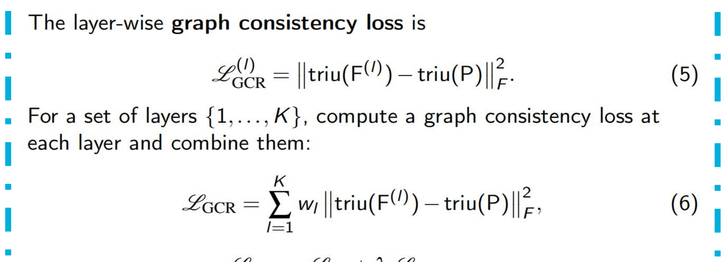
Griffith University, Queensland, Australia · Australian National University · CSIRO DATA 61 · UNSW SYDNEY

Code · Paper

NEURAL INFORMATION PROCESSING SYSTEMS

Motivation

- Deep networks learn rich features, but these features often do not match semantic class structure.
- Samples predicted as the same class may still appear far apart in feature space, hurting generalization.

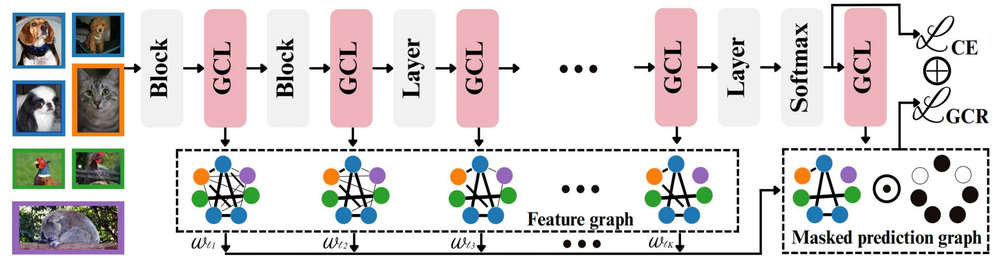


Contributions

- **GCR**: A new framework that aligns feature relational graphs with a class-aware masked prediction graph to enforce semantic consistency.
- **GCLs**: Parameter-free layers that inject multi-layer relational supervision.
- **Cross-space alignment**: Dynamically weights layers to refine semantic structure.
- **Model-agnostic improvements**: Consistent boosts in semantic coherence and generalization across diverse architectures.

Pipeline

- Insert parameter-free GCLs after any block or layer.
- Each GCL builds a feature relational graph from batch features .
- Softmax outputs form a masked prediction graph, keeping only semantically consistent pairs.
- GCLs align feature graphs to prediction graphs.
- Alignment signals are weighted and added to CE loss.



Method

We use cosine similarity with non-negative values:

$$F_{ij}^{(l)} = \mathrm{ReLU}\big(\cos(\mathrm{x}_i^{(l)}, \mathrm{x}_j^{(l)})\big), \quad i,j = 1,\ldots,n. \tag{1}$$

From the prediction logits $\mathrm{Z} = [\mathrm{z}_1^\top, \ldots, \mathrm{z}_n^\top]^\top$ of the same batch:

- apply softmax to obtain class probability vectors $\mathrm{p}_i = \mathrm{softmax}(\mathrm{z}_i)$,
- compute pairwise cosine similarity between prediction vectors:

$$S_{ij} = \mathrm{ReLU}\big(\cos(\mathrm{p}_i, \mathrm{p}_j)\big). \tag{2}$$

To focus on reliable semantic relations, we build a binary mask $\mathrm{M} \in \{0,1\}^{n\times n}$:

$$M_{ij} = \begin{cases} 1, & \text{if } y_i = y_j, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The masked prediction graph $\mathrm{P} \in \mathbb{R}^{n\times n}$ is then

$$P_{ij} = M_{ij} \odot S_{ij}, \tag{4}$$

where $\odot$ denotes elementwise multiplication.

The layer-wise **graph consistency loss** is

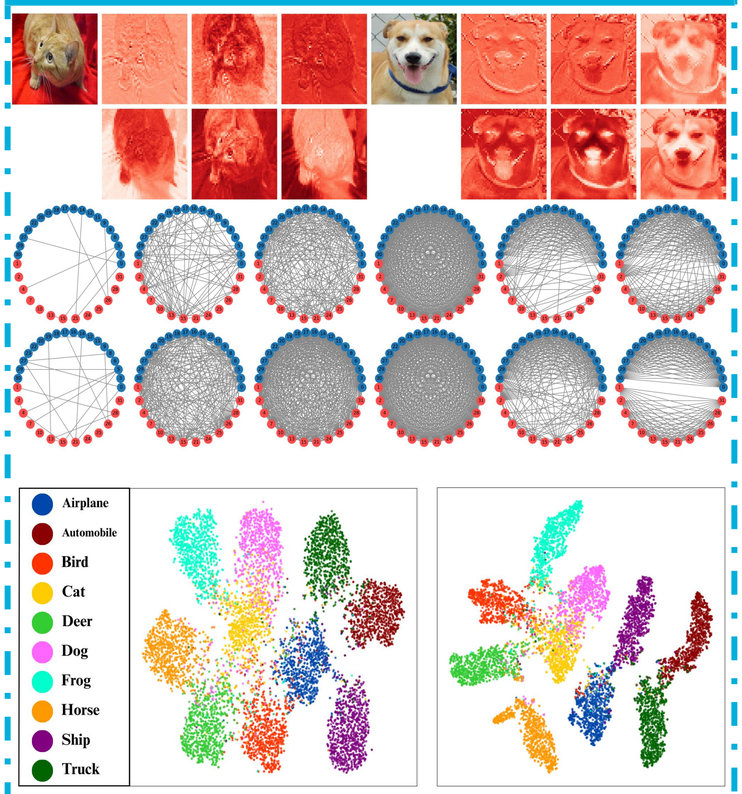
$$\mathscr{L}_{\mathrm{GCR}}^{(l)} = \big\|\mathrm{triu}(\mathrm{F}^{(l)}) - \mathrm{triu}(\mathrm{P})\big\|_F^2. \tag{5}$$

For a set of layers $\{1,\ldots,K\}$, compute a graph consistency loss at each layer and combine them:

$$\mathscr{L}_{\mathrm{GCR}} = \sum_{l=1}^{K} w_l \big\|\mathrm{triu}(\mathrm{F}^{(l)}) - \mathrm{triu}(\mathrm{P})\big\|_F^2, \tag{6}$$

$$\mathscr{L}_{\mathrm{total}} = \mathscr{L}_{\mathrm{CE}} + \lambda\, \mathscr{L}_{\mathrm{GCR}}$$

Results

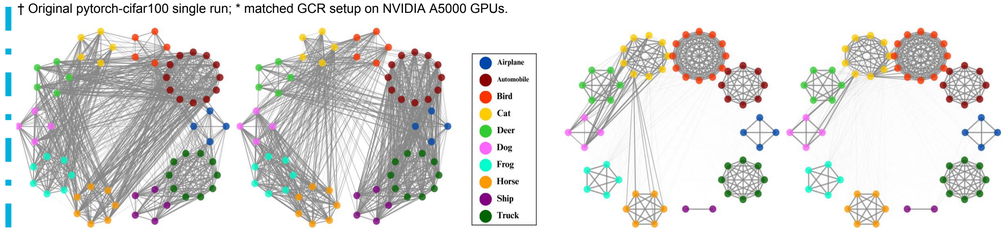


Results

	MAE	MNet	SN	SQNet	GLNet	Rx-50	Rx-101	R34	R50	R101	D121	Mean
Baseline	88.95±0.33	90.23±0.25	91.21±0.28	92.30±0.25	94.10±0.26	94.57±0.29	95.12±0.30	94.83±0.25	95.03±0.28	95.22±0.31	95.01±0.27	93.32±2.26
Early GCL	89.42±0.25	91.17±0.22	92.33±0.33	92.59±0.21	**94.89**±0.23	95.48±0.22	95.63±0.25	95.55±0.18	95.57±0.23	95.39±0.26	**95.81**±0.17	93.98±2.22
Mid GCL	**89.77**±0.22	91.15±0.18	92.58±0.19	92.40±0.20	94.82±0.21	95.47±0.19	95.39±0.24	**95.69**±0.23	95.61±0.20	**95.75**±0.17	95.51±0.22	94.01±2.15
Late GCL	89.70±0.29	**91.40**±0.19	92.36±0.21	92.80±0.19	94.88±0.19	95.35±0.28	**95.71**±0.26	**95.69**±0.19	**95.66**±0.17	95.51±0.24	95.72±0.22	**94.07**±2.14
Early+Mid	89.52±0.19	90.77±0.26	92.56±0.21	92.27±0.25	94.79±0.18	95.33±0.27	95.55±0.23	95.46±0.20	95.51±0.21	95.37±0.19	95.64±0.20	93.89±2.22
Mid+Late	89.59±0.28	91.23±0.20	**92.79**±0.20	**92.86**±0.23	94.61±0.22	**95.51**±0.19	95.38±0.27	95.45±0.18	95.33±0.26	95.52±0.14	95.70±0.19	94.00±2.09
Early+Late	89.64±0.25	91.03±0.24	92.30±0.28	92.70±0.23	94.69±0.20	95.40±0.20	95.35±0.23	95.66±0.21	95.31±0.25	95.49±0.16	95.53±0.22	93.92±2.14
Full GCL	89.55±0.23	90.99±0.18	92.48±0.19	92.65±0.20	94.57±0.21	95.50±0.19	95.34±0.20	95.48±0.17	95.62±0.18	95.38±0.21	95.51±0.20	93.92±2.15

	MAE	MNet	SN	SQNet	Rx-50	Rx-101	R34	R50	D121	Mean
Baseline†	/	65.98	70.06	69.41	77.77	77.78	76.76	77.39	77.01	/
Baseline*	64.29±0.34	67.27±0.25	71.08±0.36	70.06±0.45	79.39±0.30	79.56±0.18	77.96±0.20	79.03±0.39	79.42±0.22	74.23±6.06
Early GCL	65.05±0.29	67.45±0.21	**71.96**±0.27	70.90±0.20	79.18±0.22	79.69±0.27	77.90±0.22	79.37±0.25	79.41±0.22	74.55±5.78
Mid GCL	64.99±0.30	67.88±0.21	71.89±0.24	70.21±0.25	79.07±0.19	79.28±0.26	77.83±0.20	78.90±0.24	79.26±0.21	74.37±5.66
Late GCL	**65.54**±0.27	68.32±0.20	71.42±0.24	70.55±0.22	**79.54**±0.20	**79.83**±0.21	**78.31**±0.20	**79.42**±0.21	**79.69**±0.23	**74.74**±5.73
Early+Mid	65.23±0.31	67.62±0.24	71.50±0.28	70.47±0.19	78.90±0.18	79.25±0.20	77.41±0.19	78.58±0.24	79.22±0.20	74.28±5.56
Mid+Late	65.27±0.28	**68.33**±0.19	71.63±0.28	70.30±0.22	78.91±0.17	79.57±0.21	77.30±0.20	78.85±0.22	79.54±0.24	74.41±5.55
Early+Late	65.22±0.21	67.25±0.21	71.55±0.27	**71.03**±0.24	79.03±0.20	79.41±0.22	78.19±0.23	78.70±0.23	79.45±0.22	74.43±5.69
Full GCL	65.38±0.22	68.22±0.19	71.30±0.24	70.77±0.20	79.01±0.19	79.29±0.21	77.79±0.20	78.71±0.22	79.27±0.19	74.42±5.49

† Original pytorch-cifar100 single run; * matched GCR setup on NVIDIA A5000 GPUs.



(a) DenseNet-121 (b) With our GCLs (c) MobileNet (d) With our GCLs

The relational graphs show that adding GCLs yields cleaner, tighter class clusters with fewer cross-class links, reducing feature noise and aligning features with semantic predictions