# Graph Your Own Prompt

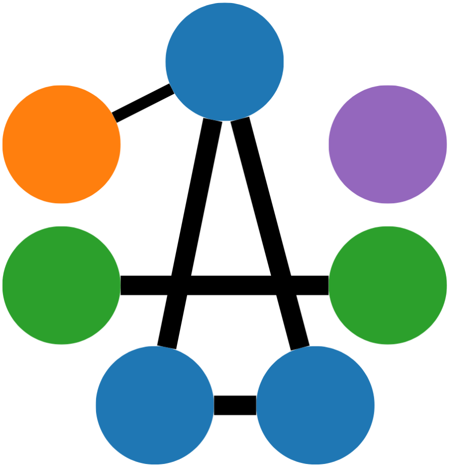
Xi Ding[1,3], Lei Wang[1,2], Piotr Koniusz[1,2,3,4], Yongsheng Gao[1]

[1]Griffith University
[2]Data61/CSIRO
[3]Australian National University
[4]University of New South Wales

NEURAL INFORMATION PROCESSING SYSTEMS

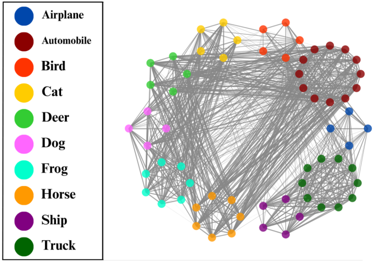
# Outline

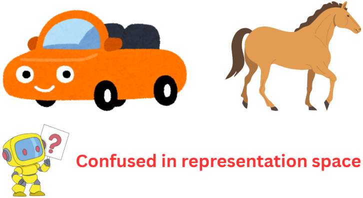
1. Motivation
2. Method
3. Experiments
4. Discussion
5. Conclusion



# Motivation: Why Structure the Feature Space?

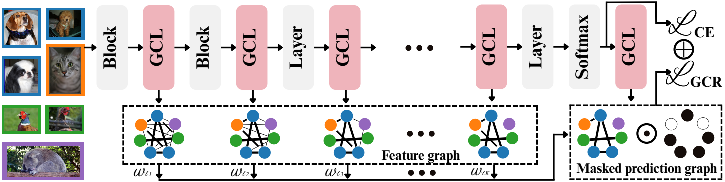
- Deep neural networks achieve strong accuracy but their internal features are often:
  - noisy and entangled,
  - misaligned with semantic class boundaries,
  - hard to interpret and fragile under distribution shifts.
- In many trained models, samples from different classes can remain close in feature space, while same-class samples may not cluster well.
- This harms generalization, robustness and interpretability, even when the final classifier performs well.



# Our Idea in One Slide

- Build relational graphs directly from:
  - intermediate features and
  - class predictions within each mini-batch.
- Use the prediction graph as a semantic reference.
- Regularize the network so that feature graphs at multiple layers align with this semantic prediction graph.
- This acts as an internal "prompt":
  - the model uses its own prediction structure to refine its feature geometry,
  - without changing the backbone architecture or adding trainable parameters.



# Overview of Graph Consistency Regularization (GCR)

- **Goal**: enforce consistency between
  - batch-level feature similarity graphs and
  - a class-aware prediction similarity graph.
- **Notation (per batch)**:
  - feature matrix at layer $l$: $\mathrm{X}^{(l)} \in \mathbb{R}^{n \times d}$, with rows $\mathrm{x}_i^{(l)}$,
  - logits: $\mathrm{Z} \in \mathbb{R}^{n \times C}$,
  - ground-truth labels: $y_1, \ldots, y_n$.
- **Key components**:
  - Graph Consistency Layers (GCLs) inserted after chosen blocks or layers,
  - a masked prediction relational graph P built from softmax outputs and labels,
  - a graph alignment loss aggregated across layers with adaptive weights.
- **Result**: semantically structured features, stronger intra-class cohesion and reduced noisy inter-class affinities.

# Graph Consistency Layer: Feature Graph

- At a chosen layer $l$, collect the batch features as a matrix:

$$\mathrm{X}^{(l)} = \begin{bmatrix} (\mathrm{x}_1^{(l)})^\top \\ \vdots \\ (\mathrm{x}_\mathrm{n}^{(\mathrm{l})})^\top \end{bmatrix} \in \mathbb{R}^{n \times d},$$

  where each row is a sample flattened into a feature vector.
- Build a feature similarity graph $\mathrm{F}^{(l)} \in \mathbb{R}^{n \times n}$:
  - nodes correspond to samples in the batch,
  - edges encode pairwise similarity between feature vectors.
- We use cosine similarity with non-negative values:

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

- This graph captures how the model currently organizes samples in feature space at that layer.

# Graph Consistency Layer: Masked Prediction Graph

- 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.
- This graph keeps intra-class semantic structure and discards possibly misleading inter-class similarities.

# Layer-wise Graph Alignment

- For each GCL at layer $l$, we align:
  - the feature graph $\mathsf{F}^{(l)}$ at that layer,
  - with the fixed masked prediction graph $\mathsf{P}$ for the batch.
- We:
  - keep only the strictly upper triangular part (undirected graph, no self-loops):

$$\mathrm{triu}(\mathsf{A}) = \text{upper triangular part of } \mathsf{A}\text{, without diagonal},$$

  - measure the discrepancy via the squared Frobenius norm.
- The layer-wise **graph consistency loss** is

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

- Small loss means feature relationships follow the semantic relationships implied by predictions; large loss means mismatch between geometry and semantics.

# Aggregating Across Layers

- GCL can be placed at multiple depths: early, middle, late, or all stages.
- For a set of layers $\{1, \ldots, K\}$, compute a graph consistency loss at each layer and combine them:

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

  where $w_l \geq 0$ are layer weights.
- **Fixed weighting** examples:

$$w_l \in \left\{ \tfrac{1}{K},\ \tfrac{l}{K},\ \left(\tfrac{l}{K}\right)^2,\ \tfrac{1+\cos(\pi l/K)}{2},\ \frac{\arccos(1-2l/K)}{\pi} \right\}.$$

- **Adaptive weighting** (graph-discrepancy-based):

$$w_l = \frac{\exp\!\big(-\|\,\mathrm{triu}(\mathrm{F}^{(l)}) - \mathrm{triu}(\mathrm{P})\|_F^2\big)}{\sum_{j=1}^{K} \exp\!\big(-\|\,\mathrm{triu}(\mathrm{F}^{(j)}) - \mathrm{triu}(\mathrm{P})\|_F^2\big)}. \tag{7}$$

# Training Objective

- The standard supervised objective is cross-entropy on the predictions:

$$\mathscr{L}_{\text{CE}}.$$

- Our total loss adds the GCR term:

$$\mathscr{L}_{\text{total}} = \mathscr{L}_{\text{CE}} + \lambda\, \mathscr{L}_{\text{GCR}}, \tag{8}$$

  where $\lambda > 0$ controls the strength of the regularization.
- Interpretation:
  - $\mathscr{L}_{\text{CE}}$ fits labels at the prediction level,
  - $\mathscr{L}_{\text{GCR}}$ aligns feature-space relations with class-aware prediction structure.
- Implementation is **portable**:
  - no new learnable parameters,
  - only involves matrix operations over the current batch,
  - can be applied to many backbones without redesigning the architecture.

# GCR as Self-Prompting

- Traditional prompting:
  - injects external tokens or instructions into the model.
- GCR can be viewed as **internal prompting**:
  - the model uses its own masked prediction graph P to generate a structural signal,
  - this signal shapes feature relationships $F^{(l)}$ across layers via the loss in (5)–(8).
- Characteristics of this internal prompt:
  - purely internal (no external tokens),
  - structural (operates on pairwise relations, not single samples),
  - semantic (class-aware masking makes the graph class-consistent).

# Experimental Setup

- **Datasets**:
    - Kaggle Cats vs. Dogs,
    - CIFAR-10 and CIFAR-100,
    - Tiny ImageNet,
    - ImageNet-1K.
- **Architectures**:
    - lightweight CNNs: MobileNet, ShuffleNet, SqueezeNet, GoogLeNet,
    - deeper CNNs: ResNet, ResNeXt, DenseNet, stochastic and squeeze-and-excitation variants,
    - vision transformers: ViT, Swin, MobileViT, CEiT, iFormer, ViG,
    - masked autoencoders.
- GCLs are inserted at early, middle, late stages, or combinations of them.

# Quantitative Results on CIFAR-10/100

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

Accuracy (%) on CIFAR-10 across models.

| | MAE | MNet | SN | SQNet | Rx-50 | Rx-101 | R34 | R50 | D121 | Mean |
|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | $64.29_{\pm 0.34}$ | $65.95_{\pm 0.25}$ | $70.11_{\pm 0.30}$ | $69.43_{\pm 0.27}$ | $77.75_{\pm 0.29}$ | $77.83_{\pm 0.30}$ | $76.82_{\pm 0.28}$ | $77.31_{\pm 0.29}$ | $77.09_{\pm 0.27}$ | $72.95_{\pm 5.50}$ |
| Early GCL | $65.05_{\pm 0.29}$ | $67.45_{\pm 0.21}$ | $\mathbf{71.96}_{\pm 0.27}$ | $\underline{70.90}_{\pm 0.20}$ | $\underline{79.18}_{\pm 0.22}$ | $\underline{79.69}_{\pm 0.27}$ | $77.90_{\pm 0.22}$ | $\underline{79.37}_{\pm 0.25}$ | $79.41_{\pm 0.22}$ | $\underline{74.55}_{\pm 5.78}$ |
| Mid GCL | $64.99_{\pm 0.30}$ | $67.88_{\pm 0.21}$ | $\underline{71.89}_{\pm 0.24}$ | $70.21_{\pm 0.25}$ | $79.07_{\pm 0.19}$ | $79.28_{\pm 0.26}$ | $77.83_{\pm 0.20}$ | $78.90_{\pm 0.24}$ | $79.26_{\pm 0.21}$ | $74.37_{\pm 5.66}$ |
| Late GCL | $\mathbf{65.54}_{\pm 0.27}$ | $\underline{68.32}_{\pm 0.20}$ | $71.42_{\pm 0.24}$ | $70.55_{\pm 0.22}$ | $\mathbf{79.54}_{\pm 0.20}$ | $\mathbf{79.83}_{\pm 0.21}$ | $\mathbf{78.31}_{\pm 0.20}$ | $\mathbf{79.42}_{\pm 0.21}$ | $\mathbf{79.69}_{\pm 0.23}$ | $\mathbf{74.74}_{\pm 5.73}$ |
| Early+Mid | $65.23_{\pm 0.31}$ | $67.62_{\pm 0.24}$ | $71.50_{\pm 0.28}$ | $70.47_{\pm 0.19}$ | $78.90_{\pm 0.18}$ | $79.25_{\pm 0.20}$ | $77.41_{\pm 0.19}$ | $78.58_{\pm 0.24}$ | $79.22_{\pm 0.20}$ | $74.28_{\pm 5.56}$ |
| Mid+Late | $65.27_{\pm 0.28}$ | $\mathbf{68.33}_{\pm 0.19}$ | $71.63_{\pm 0.28}$ | $70.30_{\pm 0.22}$ | $78.91_{\pm 0.17}$ | $79.57_{\pm 0.21}$ | $77.30_{\pm 0.20}$ | $78.85_{\pm 0.22}$ | $\underline{79.54}_{\pm 0.24}$ | $74.41_{\pm 5.55}$ |
| Early+Late | $65.22_{\pm 0.21}$ | $67.25_{\pm 0.21}$ | $71.55_{\pm 0.27}$ | $\mathbf{71.03}_{\pm 0.24}$ | $79.03_{\pm 0.20}$ | $79.41_{\pm 0.22}$ | $\underline{78.19}_{\pm 0.23}$ | $78.70_{\pm 0.23}$ | $79.45_{\pm 0.22}$ | $74.43_{\pm 5.69}$ |
| Full GCL | $\underline{65.38}_{\pm 0.22}$ | $68.22_{\pm 0.19}$ | $71.30_{\pm 0.24}$ | $70.77_{\pm 0.20}$ | $79.01_{\pm 0.19}$ | $79.29_{\pm 0.21}$ | $77.79_{\pm 0.20}$ | $78.71_{\pm 0.22}$ | $79.27_{\pm 0.19}$ | $74.42_{\pm 5.49}$ |

Accuracy (%) on CIFAR-100 across models.

# Quantitative Results on Tiny ImageNet/ImageNet-1K

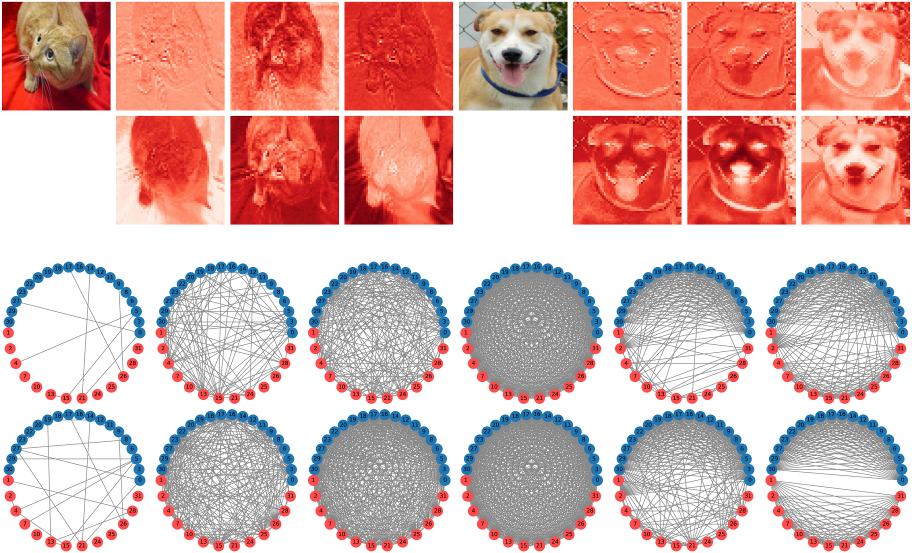
| | $\text{ViT}_{/32}$ | $\text{ViT}_{/16}$ | CeiT | $\text{MViT}_{\text{XXS}}$ | $\text{MViT}_{\text{XS}}$ | MViT | Swin | MNet | R18SD | SER18 | R34 | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | $37.79_{\pm0.35}$ | $40.05_{\pm0.33}$ | $49.95_{\pm0.29}$ | $49.28_{\pm0.29}$ | $51.58_{\pm0.27}$ | $52.68_{\pm0.27}$ | $54.27_{\pm0.25}$ | $57.81_{\pm0.25}$ | $63.49_{\pm0.26}$ | $65.65_{\pm0.24}$ | $67.51_{\pm0.25}$ | $53.64_{\pm9.62}$ |
| Early GCL | $\underline{39.02}_{\pm0.29}$ | $\underline{40.98}_{\pm0.19}$ | $\mathbf{51.22}_{\pm0.20}$ | $50.11_{\pm0.28}$ | $51.33_{\pm0.26}$ | $53.91_{\pm0.22}$ | $54.88_{\pm0.25}$ | $\mathbf{57.93}_{\pm0.21}$ | $63.81_{\pm0.19}$ | $\mathbf{66.52}_{\pm0.22}$ | $\mathbf{67.79}_{\pm0.19}$ | $\mathbf{54.32}_{\pm9.39}$ |
| Mid GCL | $38.61_{\pm0.23}$ | $40.95_{\pm0.19}$ | $50.30_{\pm0.19}$ | $49.92_{\pm0.26}$ | $51.43_{\pm0.22}$ | $53.88_{\pm0.20}$ | $55.23_{\pm0.24}$ | $57.63_{\pm0.20}$ | $64.03_{\pm0.22}$ | $65.66_{\pm0.23}$ | $67.62_{\pm0.20}$ | $54.11_{\pm9.38}$ |
| Late GCL | $37.98_{\pm0.28}$ | $40.35_{\pm0.25}$ | $\underline{50.82}_{\pm0.20}$ | $49.77_{\pm0.21}$ | $\mathbf{51.99}_{\pm0.23}$ | $\mathbf{54.10}_{\pm0.19}$ | $\underline{55.47}_{\pm0.21}$ | $57.87_{\pm0.23}$ | $63.79_{\pm0.19}$ | $65.85_{\pm0.25}$ | $67.61_{\pm0.19}$ | $54.18_{\pm9.56}$ |
| Early+Mid | $\mathbf{39.08}_{\pm0.25}$ | $\mathbf{41.26}_{\pm0.18}$ | $50.25_{\pm0.25}$ | $49.73_{\pm0.22}$ | $51.57_{\pm0.19}$ | $53.91_{\pm0.23}$ | $54.95_{\pm0.19}$ | $57.49_{\pm0.19}$ | $\underline{64.18}_{\pm0.20}$ | $65.86_{\pm0.24}$ | $67.74_{\pm0.23}$ | $54.18_{\pm9.32}$ |
| Mid+Late | $38.44_{\pm0.18}$ | $40.52_{\pm0.28}$ | $50.09_{\pm0.25}$ | $\mathbf{50.55}_{\pm0.18}$ | $51.48_{\pm0.21}$ | $53.90_{\pm0.20}$ | $\mathbf{55.62}_{\pm0.23}$ | $57.65_{\pm0.21}$ | $\mathbf{64.29}_{\pm0.17}$ | $65.95_{\pm0.19}$ | $67.58_{\pm0.21}$ | $\underline{54.19}_{\pm9.52}$ |
| Early+Late | $38.34_{\pm0.23}$ | $40.71_{\pm0.21}$ | $50.70_{\pm0.25}$ | $\underline{50.23}_{\pm0.20}$ | $51.36_{\pm0.18}$ | $53.57_{\pm0.21}$ | $54.89_{\pm0.21}$ | $\underline{57.93}_{\pm0.19}$ | $63.88_{\pm0.19}$ | $65.83_{\pm0.17}$ | $\underline{67.75}_{\pm0.25}$ | $54.11_{\pm9.47}$ |
| Full GCL | $38.38_{\pm0.22}$ | $40.80_{\pm0.18}$ | $49.92_{\pm0.20}$ | $50.16_{\pm0.17}$ | $\underline{51.87}_{\pm0.19}$ | $\underline{54.01}_{\pm0.19}$ | $54.87_{\pm0.19}$ | $57.64_{\pm0.20}$ | $64.10_{\pm0.19}$ | $\underline{66.01}_{\pm0.15}$ | $67.66_{\pm0.18}$ | $54.13_{\pm9.49}$ |

Accuracy (%) on Tiny ImageNet across models.

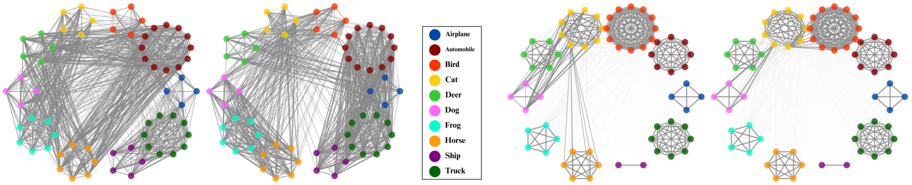
| Method | iFormer-S | iFormer-B | ViT-B/16 | ViG-B |
|---|---|---|---|---|
| Baseline | $83.4_{\pm0.40}$ | $84.6_{\pm0.45}$ | $74.3_{\pm0.51}$ | $82.3_{\pm0.42}$ |
| Early GCL | $83.8_{\pm0.31}$ | $85.0_{\pm0.40}$ | $74.7_{\pm0.44}$ | $82.8_{\pm0.35}$ |
| Mid GCL | $83.8_{\pm0.39}$ | $85.5_{\pm0.33}$ | $75.2_{\pm0.36}$ | $83.0_{\pm0.34}$ |
| Late GCL | $\underline{84.5}_{\pm0.29}$ | $\mathbf{86.1}_{\pm0.30}$ | $\mathbf{75.8}_{\pm0.33}$ | $\mathbf{84.0}_{\pm0.30}$ |
| Early + Mid | $84.3_{\pm0.33}$ | $\underline{85.9}_{\pm0.38}$ | $\underline{75.6}_{\pm0.41}$ | $83.7_{\pm0.33}$ |
| Mid + Late | $\mathbf{84.8}_{\pm0.28}$ | $\underline{85.9}_{\pm0.37}$ | $\underline{75.6}_{\pm0.34}$ | $\underline{83.9}_{\pm0.30}$ |
| Early + Late | $\underline{84.5}_{\pm0.30}$ | $85.2_{\pm0.28}$ | $74.9_{\pm0.33}$ | $83.5_{\pm0.29}$ |
| Full GCL | $84.3_{\pm0.29}$ | $85.8_{\pm0.26}$ | $75.5_{\pm0.30}$ | $83.6_{\pm0.27}$ |

Accuracy (%) on ImageNet-1K across models.

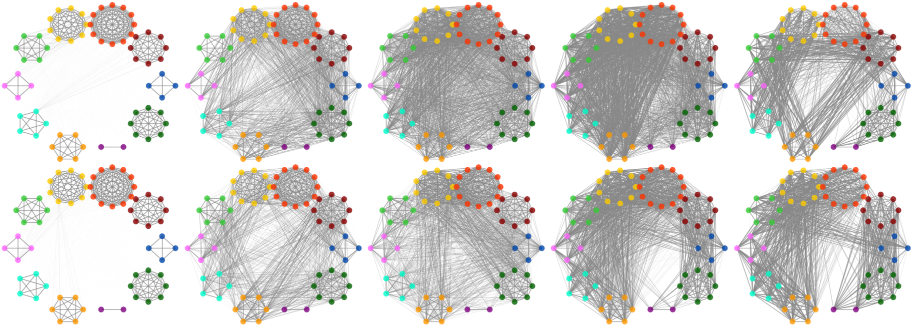
# Qualitative Results on Kaggle Cats vs. Dogs



# Qualitative Results on CIFAR-10

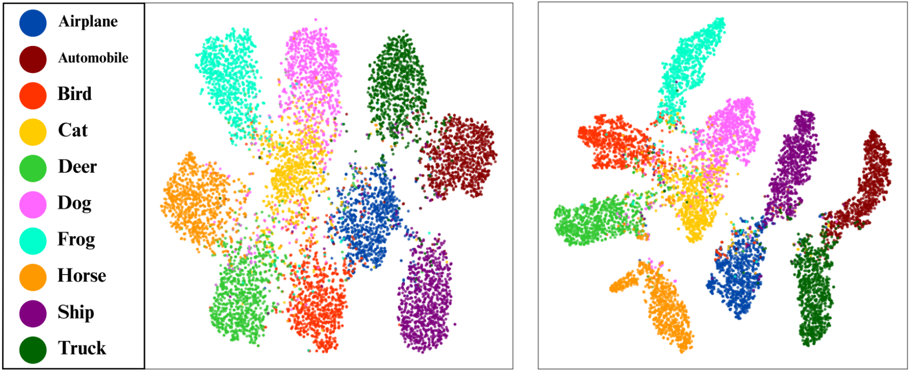



(a) DenseNet-121 (b) With our GCLs (c) MobileNet (d) With our GCLs



(a) ShuffleNet (b) GoogLeNet (c) ResNet-50 (d) ResNet-101 (e) DenseNet-121

# Qualitative Results on CIFAR-10




Left is the baseline (ShuffleNet), and the right is GCL-augmented.
GCL-augmented yields more compact intra-class clusters and better inter-class separation.

# Where to Place & How to Weight GCLs?

| | E | M | L | E+M | M+L | E+L | Full |
|---|---|---|---|---|---|---|---|
| Linear | 0.55 | 0.45 | 0.48 | 0.31 | 0.34 | 0.36 | 0.27 |
| Sqrt | 0.44 | 0.41 | 0.42 | 0.21 | 0.33 | 0.26 | 0.19 |
| Squared | 0.55 | 0.46 | 0.49 | 0.51 | 0.49 | 0.47 | 0.40 |
| Equal | 0.49 | 0.58 | 0.56 | 0.30 | 0.47 | 0.45 | 0.45 |
| Arccos | 0.51 | 0.53 | 0.53 | 0.34 | 0.40 | 0.40 | 0.24 |
| Cosine | 0.39 | 0.44 | 0.57 | 0.23 | 0.45 | 0.39 | 0.19 |
| Adaptive | 0.48 | 0.56 | 0.63 | 0.45 | 0.61 | 0.53 | 0.51 |

CIFAR-10

| | E | M | L | E+M | M+L | E+L | Full |
|---|---|---|---|---|---|---|---|
| Linear | 1.40 | 1.20 | 1.30 | 0.80 | 0.96 | 0.95 | 0.77 |
| Sqrt | 1.15 | 1.02 | 1.08 | 0.66 | 0.75 | 0.79 | 0.40 |
| Squared | 1.47 | 1.16 | 1.14 | 1.12 | 0.90 | 1.14 | 0.97 |
| Equal | 1.10 | 1.11 | 1.24 | 1.14 | 1.37 | 1.11 | 1.26 |
| Arccos | 1.38 | 1.17 | 0.98 | 0.85 | 0.76 | 0.92 | 0.53 |
| Cosine | 1.17 | 1.23 | 1.52 | 0.60 | 1.14 | 1.22 | 0.67 |
| Adaptive | 1.45 | 1.30 | 1.57 | 1.27 | 1.23 | 1.42 | 1.30 |

CIFAR-100

- **Placement:**
  - Late GCLs yield the largest gains.
  - Early GCLs help stabilize feature learning and reduce noise propagation.
  - Combining late with other stages is often beneficial.
  - Full GCL not always best (too much regularization may hurt).
- **Weighting:**
  - Adaptive weighting works best (focuses on misaligned layers).
  - Fixed schemes: squared / equal $>$ linear / square root.

# What Does GCR Buy Us?

- **Semantic feature geometry**:
  - features are organized to respect class-aware relationships in prediction space,
  - intra-class cohesion increases, inter-class confusion decreases.
- **Better generalization**:
  - graph-based regularization shrinks the effective hypothesis space,
  - empirical results show consistent gains across datasets and architectures.
- **Interpretability**:
  - relational graphs become easier to interpret,
  - feature maps highlight semantically meaningful regions.

# GCR as a Bridge Between Paradigms

- **Graph-based learning**:
  - GCR builds graphs dynamically within each batch,
  - no need for persistent global graphs or specialized graph neural networks.
- **Contrastive-style supervision**:
  - graph alignment can be seen as a soft form of contrast,
  - encourages similar samples to be close and dissimilar samples to be separated,
  - without explicit positive and negative sampling.
- **Prompting and self-conditioning**:
  - prediction graphs act as continuous, internal prompts that shape representation learning.

# When Is GCR Most Helpful?

- Datasets with many classes or fine-grained distinctions:
  - benefit from stronger control over feature geometry,
  - GCR helps separate visually similar but semantically different classes.
- Architectures with limited capacity:
  - lightweight models gain substantial accuracy from the extra structural supervision.
- Deeper backbones:
  - late layers encode rich semantics, and graph alignment at these layers is especially effective.

# Limitations and Practical Considerations

- Requires labels to build the class-aware mask:
  - the current formulation is fully supervised,
  - extensions to semi-supervised or self-supervised settings remain open work.
- Graph construction is batch-based:
  - complexity grows with batch size,
  - in practice, the cost is manageable due to efficient matrix operations.
- Strong reliance on prediction quality:
  - very early in training predictions may be noisy,
  - however, the class-aware mask mitigates the influence of unreliable inter-class similarities.

# Summary

- Introduced **Graph Consistency Regularization (GCR)**:
  - a parameter-free, model-agnostic framework,
  - aligns batch-level feature graphs with a class-aware prediction graph.
- Proposed **Graph Consistency Layers (GCLs)**:
  - lightweight modules that can be inserted at arbitrary depths,
  - enforce multi-layer structural supervision with flexible weighting.
- Demonstrated:
  - improved semantic structure of feature spaces,
  - stronger intra-class cohesion and clearer class boundaries,
  - consistent accuracy gains across many models and datasets.

# Outlook and Future Work

- Extend GCR beyond image classification:
  - semantic and instance segmentation,
  - retrieval and metric learning tasks,
  - multi-modal settings such as vision-language models.
- Combine GCR with self-supervised and semi-supervised learning:
  - use pseudo-labels or clustering to build masked prediction graphs without full supervision.
- Explore more advanced graph constructions:
  - adaptive sparsification,
  - other similarity measures and masking strategies,
  - alternative ways to incorporate temporal or multi-view structure.

# Thank You

Paper

Code

Scan the QR codes to read our paper and code.