Graph Condensation for Graph Neural Networks

Wei Jin, Lingxiao Zhao, Shichang Zhang, Yozen Liu, Jiliang Tang, Neil Shah

ICLR 2022
Data as Graphs

Social Graphs

Transportation Graphs

Brain Graphs

Web Graphs

Molecular Graphs

Gene Graphs
Graph Neural Networks

\[ G = (A, X) \]

**Graph Convolutions**

**Activation Function**

**Representations**
- Graph-level
- Node-level

\[ \mathbf{G} = (\mathbf{A}, \mathbf{X}) \]
Applications of GNNs

• Node-level
  • Friend/product recommendation
  • User interest prediction
  • Fraud Detection
  • ....

• Graph-level
  • Molecules property prediction
  • Drug discovery
  • ....
Challenges: Learning with Large-Scale Graphs

- Model training time
  - Neural architecture search
  - Continual learning

- Storage

- Visualization

Is it possible to significantly reduce the graph size while providing sufficient information to well-train GNNs?
Our Problem: Graph Condensation

• Problem definition

Given graph $\mathcal{T} = (\mathbf{A}, \mathbf{X}, \mathbf{Y})$, we aim to learn a graph $\mathcal{S} = (\mathbf{A}', \mathbf{X}', \mathbf{Y}')$ with fewer nodes such that a GNN trained on $\mathcal{S}$ can achieve comparable performance to one trained on $\mathcal{T}$.

$$\min_{\mathcal{S}} \mathcal{L} \left( \text{GNN}_{\theta_{\mathcal{S}}} (\mathbf{A}, \mathbf{X}), \mathbf{Y} \right)$$

Minimize loss on training graph

s.t. $\theta_{\mathcal{S}} = \arg \min_{\theta} \mathcal{L} (\text{GNN}_{\theta}(\mathbf{A}', \mathbf{X}'), \mathbf{Y}')$

via learned parameters on condensed graph
Our Problem: Graph Condensation

- Problem definition

Given graph $\mathcal{T} = (A, X, Y)$, we aim to learn a graph $\mathcal{S} = (A', X', Y')$ with fewer nodes such that a GNN trained on $\mathcal{S}$ can achieve comparable performance to one trained on $\mathcal{T}$.
Graph Condensation via Gradient Matching

• Directly optimizing the bi-level problem is hard!

• Instead, we aim to learn $S$ that can lead to a similar solution as $T$:

$$\min_S \left[ \sum_{t=0}^{T-1} D \left( \theta_t^S, \theta_t^T \right) \right]$$

with

$$\theta_{t+1}^S = \text{opt}_\theta \left( \mathcal{L} \left( \text{GNN}_{\theta_t^S}(A', X'), Y' \right) \right)$$

and

$$\theta_{t+1}^T = \text{opt}_\theta \left( \mathcal{L} \left( \text{GNN}_{\theta_t^T}(A, X), Y \right) \right)$$

- Params from condensed graph
- Params from original graph

• We can convert the problem to matching gradients\[1\]:

$$\min_S \left[ \sum_{t=0}^{T-1} D \left( \nabla_\theta \mathcal{L} \left( \text{GNN}_{\theta_t}(A', X'), Y' \right), \nabla_\theta \mathcal{L} \left( \text{GNN}_{\theta_t}(A, X), Y \right) \right) \right]$$

Minimize distance between gradients at each step

\[1\] Dataset Condensation with Gradient Matching. ICLR 2021.
GCond Framework: Overview

• Learning $S$ can lead to a similar solution as $T$ through the following gradient matching scheme

$$\min_S \left[ \sum_{t=0}^{T-1} D(\nabla_\theta \mathcal{L}(\text{GNN}_{\theta_t}(A', X'), Y'), \nabla_\theta \mathcal{L}(\text{GNN}_{\theta_t}(A, X), Y)) \right]$$
GCond Framework: Overview

• Learning $\mathcal{S}$ can lead to a similar solution as $\mathcal{T}$ through the following gradient matching scheme

$$\min_{\mathcal{S}} \left[ \sum_{t=0}^{T-1} D(\nabla_{\theta} \mathcal{L}(\text{GNN}_{\theta_t}(A', X'), Y'), \nabla_{\theta} \mathcal{L}(\text{GNN}_{\theta_t}(A, X), Y)) \right]$$

How can we parametrize the learning of $\mathcal{S}$?
GCond Framework: Overview

• We fix $\mathbf{Y}'$ w.r.t class distribution to simplify the problem

• We model $\mathbf{X}'$ as free parameters

• We use an MLP to model condensed node relations

$$A' = g_\Phi(\mathbf{X}') ,$$

with $A'_{i,j} = \text{Sigmoid}\left( \frac{\text{MLP}_\Phi([x'_i; x'_j]) + \text{MLP}_\Phi([x'_j; x'_i])}{2} \right)$
GCond Framework: Graph Modeling

• We use an MLP to model condensed node relations

\[ A' = g_{\Phi}(X'), \]

with \[ A'_{i,j} = \text{Sigmoid} \left( \frac{\text{MLP}_{\Phi}([x'_i; x'_j]) + \text{MLP}_{\Phi}([x'_j; x'_i])}{2} \right) \]

• This has crucial benefits:

  • # parameters independent of input graph size
  • MLP_{\Phi} can be used to inductively “grow” condensed graph
  • empirically better than modeling \( A' \) as free parameters
GCond Framework: Other Details

- Graph sampling on original graph
  - Use mini-batch training, since training on the original graph can be expensive.

- Sparsification of the condensed graph
  \[ A' = \text{ReLU}(g_\Phi(X') - \delta) \]

- Alternating optimization
  - Alternatively update \( g_\Phi \) and \( X' \) to ease optimization
Evaluation of Condensed Graphs

• Does condensation preserve original graph information?

• Can condensed graphs be used to train different GNNs?

• What do the condensed graphs look like?

• Can graph condensation help neural architecture search?
## Performance with Condensed Graphs

- Condensed graphs can provide sufficient information to train GNNs.

| Dataset    | Ratio ($r$) | Baselines                      | Proposed                     | Whole Dataset |
|------------|-------------|-------------------------------|------------------------------|---------------|
|            |             | Random ($A', X'$)             | Herding ($A', X'$)           | Coarsening ($A', X'$) | DC-Graph ($X$') | GCOND-X ($X'$) | GCOND ($A', X'$) |               |
| Citeeseer  | 0.9%        | 54.4±4.4                      | 57.1±1.5                     | 52.4±2.8      | 52.2±0.4       | 66.8±1.5       | **71.4±0.8**    | 70.5±1.2      |
|            | 1.8%        | 64.2±1.7                      | 66.7±1.0                     | 64.3±1.0      | 59.0±0.5       | 66.9±0.9       | **70.6±0.9**    | 71.7±0.1      |
|            | 3.6%        | 69.1±0.1                      | 69.0±0.1                     | 69.1±0.1      | 65.3±0.5       | 66.3±1.5       | **69.4±1.4**    | **69.8±1.4**  |
| Cora       | 1.3%        | 63.6±3.7                      | 67.0±1.3                     | 64.0±2.3      | 31.2±0.2       | 67.3±1.9       | 75.9±1.2       | **79.8±1.3**  |
|            | 2.6%        | 72.8±1.1                      | 73.4±1.0                     | 73.2±1.2      | 65.2±0.6       | 67.6±3.5       | 75.7±0.9       | **80.1±0.6**  | 81.2±0.2      |
|            | 5.2%        | 76.8±0.1                      | 76.8±0.1                     | 76.7±0.1      | 70.6±0.1       | 67.7±2.2       | 76.0±0.9       | **79.3±0.3**  |
| Ogbn-arxiv | 0.05%       | 47.1±3.9                      | 52.4±1.8                     | 47.2±3.0      | 35.4±0.3       | 58.6±0.4       | **61.3±0.5**    | 59.2±1.1      |
|            | 0.25%       | 57.3±1.1                      | 58.6±1.2                     | 56.8±0.8      | 43.5±0.2       | 59.9±0.3       | **64.2±0.4**    | 63.2±0.3      | 71.4±0.1      |
|            | 0.5%        | 60.0±0.9                      | 60.4±0.8                     | 60.3±0.4      | 50.4±0.1       | 59.5±0.3       | 63.1±0.5       | **64.0±0.4**  |
| Flickr     | 0.1%        | 41.8±2.0                      | 42.5±1.8                     | 42.0±0.7      | 41.9±0.2       | 46.3±0.2       | 45.9±0.1       | **46.5±0.4**  |
|            | 0.5%        | 44.0±0.4                      | 43.9±0.9                     | 43.2±0.1      | 44.5±0.1       | 45.9±0.1       | 45.0±0.2       | **47.1±0.1**  | 47.2±0.1      |
|            | 1%          | 44.6±0.2                      | 44.4±0.6                     | 44.1±0.4      | 44.6±0.1       | 45.8±0.1       | 45.0±0.1       | **47.1±0.1**  |
| Reddit     | 0.05%       | 46.1±4.4                      | 53.1±2.5                     | 46.6±2.3      | 40.9±0.5       | 88.2±0.2       | **88.4±0.4**    | 88.0±1.8      |
|            | 0.1%        | 58.0±2.2                      | 62.7±1.0                     | 53.0±3.3      | 42.8±0.8       | 89.5±0.1       | **89.6±0.7**    | 93.9±0.0      |
|            | 0.2%        | 66.3±1.9                      | 71.0±1.6                     | 58.5±2.1      | 47.4±0.9       | **90.5±1.2**   | 88.8±0.4       | 90.1±0.5      |
Performance on Various GNNs

• Condensed graphs can well-train different GNNs.

| Methods   | Data  | MLP | GAT | APPNP | Cheby | GCN | SAGE | SGC | Avg. | Whole Dataset |
|-----------|-------|-----|-----|-------|-------|-----|------|-----|------|---------------|
| Citeseer  |       |     |     |       |       |     |      |     |      | 71.7±0.1      |
| $r = 1.8\%$ |       |     |     |       |       |     |      |     |      |               |
| DC-Graph  | $X'$  | 66.2| -   | 66.4  | 64.9  | 66.2| 65.9 | 69.6| 66.6 |               |
| GCOND-X   | $X'$  | 69.6| -   | 69.7  | 70.6  | 69.7| 69.2 | 71.6| 70.2 |               |
| GCOND     | $A', X'$ | 63.9| 55.4| 69.6  | 68.3  | 70.5| 66.2 | 70.3| 69.0 |               |
| Cora      |       |     |     |       |       |     |      |     |      | 81.2±0.2      |
| $r = 2.6\%$ |       |     |     |       |       |     |      |     |      |               |
| DC-Graph  | $X'$  | 67.2| -   | 67.1  | 67.7  | 67.9| 66.2 | 72.8| 68.3 |               |
| GCOND-X   | $X'$  | 76.0| -   | 77.0  | 74.1  | 75.3| 76.0 | 76.1| 75.7 |               |
| GCOND     | $A', X'$ | 73.1| 66.2| 78.5  | 76.0  | 80.1| 78.2 | 79.3| 78.4 |               |
| Ogbn-arxiv|       |     |     |       |       |     |      |     |      | 71.4±0.1      |
| $r = 0.25\%$ |       |     |     |       |       |     |      |     |      |               |
| DC-Graph  | $X'$  | 59.9| -   | 60.0  | 55.7  | 59.8| 60.0 | 60.4| 59.2 |               |
| GCOND-X   | $X'$  | 64.1| -   | 61.5  | 59.5  | 64.2| 64.4 | 64.7| 62.9 |               |
| GCOND     | $A', X'$ | 62.2| 60.0| 63.4  | 54.9  | 63.2| 62.6 | 63.7| 61.6 |               |
| Flickr    |       |     |     |       |       |     |      |     |      | 47.2±0.1      |
| $r = 0.5\%$ |       |     |     |       |       |     |      |     |      |               |
| DC-Graph  | $X'$  | 43.1| -   | 45.7  | 43.8  | 45.9| 45.8 | 45.6| 45.4 |               |
| GCOND-X   | $X'$  | 42.1| -   | 44.6  | 42.3  | 45.0| 44.7 | 44.4| 44.2 |               |
| GCOND     | $A', X'$ | 44.8| 40.1| 45.9  | 42.8  | 47.1| 46.2 | 46.1| 45.6 |               |
| Reddit    |       |     |     |       |       |     |      |     |      | 93.9±0.0      |
| $r = 0.1\%$ |       |     |     |       |       |     |      |     |      |               |
| DC-Graph  | $X'$  | 50.3| -   | 81.2  | 77.5  | 89.5| 89.7 | 90.5| 85.7 |               |
| GCOND-X   | $X'$  | 40.1| -   | 78.7  | 74.0  | 89.3| 89.3 | 91.0| 84.5 |               |
| GCOND     | $A', X'$ | 42.5| 60.2| 87.8  | 75.5  | 89.4| 89.1 | 89.6| 86.3 |               |

GCN performance
Cross-Architecture Experiments

- Graphs condensed by different GNNs all show strong transfer performance on other architecture

(a) Cora, $r=2.6\%$

| C\T     | APPNP  | Cheby  | GCN    | SAGE   | SGC    |
|---------|--------|--------|--------|--------|--------|
| APPNP   | 72.1±2.6 | 60.8±6.4 | 73.5±2.4 | 72.3±3.5 | 73.1±3.1 |
| Cheby   | 75.3±2.9 | 71.8±1.1 | 76.8±2.1 | 76.4±2.0 | 75.5±3.5 |
| GCN     | 69.8±4.0 | 53.2±3.4 | 70.6±3.7 | 60.2±1.9 | 68.7±5.4 |
| SAGE    | 77.1±1.1 | 69.3±1.7 | 77.0±0.7 | 76.1±0.7 | 77.7±1.8 |
| SGC     | 78.5±1.0 | 76.0±1.1 | 80.1±0.6 | 78.2±0.9 | 79.3±0.7 |

C: model used for condensation
T: model used for test
Statistics of Condensed Graphs

• Condensed graphs require much less storage.

• Homophily information is often preserved (e.g. Cora, Citeseer and Flickr).

Table 5: Comparison between condensed graphs and original graphs. The condensed graphs have fewer nodes and are more dense.

|            | Citeseer, \( r=0.9\% \) | Cora, \( r=1.3\% \) | Ogbn-arxiv, \( r=0.25\% \) | Flickr, \( r=0.5\% \) | Reddit, \( r=0.1\% \) |
|------------|-------------------------|----------------------|-----------------------------|-------------------------|-------------------------|
|            | Whole  | GCOND | Whole  | GCOND | Whole  | GCOND | Whole  | GCOND | Whole  | GCOND |
| Accuracy   | 70.7   | 70.5   | 81.5   | 79.8   | 71.4   | 63.2   | 47.1   | 47.1   | 94.1   | 89.4   |
| #Nodes     | 3,327  | 60     | 2,708  | 70     | 169,343| 454    | 44,625 | 223    | 153,932| 153    |
| #Edges     | 4,732  | 1,454  | 5,429  | 2,128  | 1,166,243| 3,354  | 218,140| 3,788  | 10,753,238| 301   |
| Sparsity   | 0.09\% | 80.78\%| 0.15\% | 86.86\%| 0.01\% | 3.25\% | 0.02\% | 15.23\%| 0.09\% | 2.57\% |
| Homophily  | 0.74   | 0.65   | 0.81   | 0.79   | 0.65   | 0.07   | 0.33   | 0.28   | 0.78   | 0.04   |
| Storage    | 47.1 MB| 0.9 MB | 14.9 MB| 0.4 MB | 100.4 MB| 0.3 MB | 86.8 MB| 0.5 MB | 435.5 MB| 0.4 MB |
Visualization of Condensed Graphs

• Condensed graphs require much less storage.

• Homophily information is often preserved (e.g. Cora, Citeseer and Flickr).

(a) Cora, $r=2.5\%$  (b) Citeseer, $r=1.8\%$  (c) Arxiv, $r=0.05\%$  (d) Flickr, $r=0.1\%$  (e) Reddit, $r=0.1\%$
Neural Architecture Search

- Setup:

We focus on APPNP instead of GCN since the architecture of APPNP involves more hyperparameters regarding its architecture setup. The detailed search space is shown as follows:

(a) **Number of propagation** $K$: we search the number of propagation $K$ in the range of \( \{2, 4, 6, 8, 10\} \).

(b) **Residual coefficient** $\alpha$: for the residual coefficient in APPNP, we search it in the range of \( \{0.1, 0.2\} \).

(c) **Hidden dimension**: We collect the set of dimensions that are widely adopted by existing work as the candidates, i.e., \( \{16,32,64,128,256,512\} \).

(d) **Activation function**: The set of available activation functions is listed as follows: \{Sigmoid, Tanh, ReLU, Linear, Softplus, LeakyReLU, ReLU6, ELU\}

In total, there are 480 architectures to be searched.
Neural Architecture Search

- GCond has strong +ve correlation between condensed val and original val performance.

Table 10: Neural Architecture Search. Methods are compared in validation accuracy correlation (with the original validation accuracy) obtained by searched architecture.

|          | Random | Herding | GCOND |
|----------|--------|---------|-------|
| Cora     | 0.40   | 0.21    | 0.76  |
| Citeseer | 0.56   | 0.29    | 0.79  |
| Ogbn-arxiv | 0.63 | 0.6     | 0.64  |
Conclusion

• We make the first (neural) attempt to condense a large-real graph into a small-synthetic graph.

• We propose a practical learning techniques and choices which impact condensation performance.

• We show GCond is parsimonious and versatile.

• We showcase the promise of condensation in NAS applications.
One-step Gradient Matching

• Condensing Graphs via One-Step Gradient Matching. KDD 2022

• Perform gradient matching for only one single step without training the network weights
One-step Gradient Matching

• Condensing Graphs via One-Step Gradient Matching. KDD 2022
• Perform gradient matching for **only one single step without training the network weights**
• The smallest gap between the resulted loss (by training on synthetic graphs) and the optimal loss has an up

\[
\min_{t=0,1,\ldots,T-1} \ell_T (\theta_t) - \ell_T (\theta^*) \leq \sum_{t=0}^{T-1} \frac{\sqrt{2M}}{T} \| \nabla_{\theta} \ell_T (\theta_t) - \nabla_{\theta} \ell_S (\theta_t) \| \\
+ \frac{3M}{2\sqrt{T}} \cdot \frac{C - 1}{CN'} \| A'KX' \| 
\]  

(Theorem 2. When we use a K-layer SGC as the model used in condensation, i.e., \( f_{\theta}(A, X, \theta) = A^KXW \) with \( \theta = W \) and assume that all network parameters satisfy \( \| \theta \|^2 \leq M^2 (M > 0) \), we have)
One-step Gradient Matching

- Condensing Graphs via One-Step Gradient Matching. KDD 2022
- Perform gradient matching for *only one single step without training the network weights*

Table 3: Node classification accuracy (%) comparison. The numbers in parentheses indicate the running time for 100 epochs and $r$ indicates the ratio of number of nodes in the condensed graph to that in the original graph.

|          | Cora, $r=2.6\%$ | Citeseer, $r=1.8\%$ | Pubmed, $r=0.3\%$ | Arxiv, $r=0.25\%$ | Flickr, $r=0.1\%$ |
|----------|------------------|----------------------|-------------------|-------------------|-------------------|
| $GCond$  | 80.1 (75.9s)     | 70.6 (71.8s)         | 77.9 (51.7s)      | 59.2 (494.3s)     | 46.5 (51.9s)      |
| $DosCond$| 80.0 (3.5s)      | 71.0 (2.8s)          | 76.0 (1.3s)       | 59.0 (32.9s)      | 46.1 (14.3s)      |
| Whole Dataset | 81.5 | 71.7 | 79.3 | 71.4 | 47.2 |
THANK YOU

My webpage: [http://cse.msu.edu/~jinwei2/](http://cse.msu.edu/~jinwei2/)
Paper: [https://openreview.net/pdf?id=WLEx3Jo4QaB](https://openreview.net/pdf?id=WLEx3Jo4QaB)
Code: [https://github.com/ChandlerBang/GCond](https://github.com/ChandlerBang/GCond)