A. Appendix

```python
#M as ascent steps, alpha as ascent step size
#X denotes input node features, y denotes labels
def flag(gnn, X, y, optimizer, criterion, M, alpha) :
    gnn.train()
    optimizer.zero_grad()

    pert = torch.FloatTensor(*X.shape).uniform_(-alpha, alpha)
    pert.requires_grad_()
    out = gnn(X+pert)
    loss = criterion(out, y)/M

    for _ in range(M-1):
        loss.backward()
        pert_data = pert.detach() + alpha*torch.sign(pert.grad.detach())
        pert.data = pert_data.data
        pert.grad[:] = 0
        out = gnn(X+pert)
        loss = criterion(out, y)/M

    loss.backward()
    optimizer.step()
```

Figure 5. An abstract PyTorch Implementation of our method.

We summarize implementation details and selected hyperparameters in this section. Note that for ALL of our method, we fix the ascent step number M to 3 for simplicity. We leave more thorough step number search for future research. Experiments are done on hardware with Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, and 128GB of RAM. If without mentioning, we use a single GeForce RTX 2080 Ti (11GB GPU memory). To highlight, for fair comparisons, we do not modify model architectures nor optimizing algorithms.

A.1. Node Classification

A.1.1 ogbn-products

MLP: perturbation step size α=2e-02, only labeled nodes are used in the training phase.

GraphSAGE: labeled perturbation step size α_l=8e-03, α_u/α_l=2, and neighbor sampling is used for scalable training.

GAT: labeled perturbation step size α_l=5e-03, α_u/α_l=2, and neighbor sampling is used for scalable training.

DeeperGCN: labeled perturbation step size α_l=5e-03, α_u/α_l=2, and the model is trained on NVIDIA Tesla V100 (32GB GPU memory).

A.1.2 ogbn-proteins

GCN: labeled perturbation step size α_l=1e-03, α_u/α_l=1, and the model is trained in the full-batch manner.

GraphSAGE: labeled perturbation step size α_l=1e-03, α_u/α_l=1, and the model is trained in the full-batch manner.

DeeperGCN: labeled perturbation step size α_l=8e-03, α_u/α_l=1, and the model is trained on NVIDIA Tesla V100 (32GB GPU memory).

A.1.3 ogbn-arxiv

MLP: perturbation step size α=2e-03, only labeled nodes are used in the training phase.

GCN: labeled perturbation step size α_l=1e-03, α_u/α_l=1, and the model is trained in the full-batch manner.

GraphSAGE: labeled perturbation step size α_l=1e-03, α_u/α_l=1, and the model is trained in the full-batch manner.

GAT: labeled perturbation step size α_l=1e-03, α_u/α_l=2, and the model is trained in the full-batch manner.

DeeperGCN: labeled perturbation step size α_l=8e-03, α_u/α_l=1, and the model is trained on NVIDIA Tesla V100 (32GB GPU memory).

A.1.4 ogbn-mag

R-GCN: labeled perturbation step size α_l=1e-04, α_u/α_l=1, and the model is trained with neighbor sampling for scalability.

A.2. Link Prediction

A.2.1 ogbl-ddi

GCN perturbation step size α=3e-03, and **GraphSAGE** perturbation step size α_l=3e-03. Models are both trained in the full-batch manner. During each gradient ascent loop negative edges are resampled for computing negative losses.

A.2.2 ogbl-collab

GCN perturbation step size α=3e-03, and **GraphSAGE** perturbation step size α_l=3e-03. Models are both trained in the full-batch manner. During each gradient ascent loop negative edges are resampled for computing negative losses.

A.3. Graph Classification

A.3.1 ogbg-molhiv

GCN: perturbation step size α=1e-02, when virtual node is added we use a smaller α=1e-03.

GIN: perturbation step size α=5e-03, when virtual node is added we use a smaller α=1e-03.

DeeperGCN: perturbation step size α=1e-02, and the model is trained on NVIDIA Tesla V100 (32GB GPU memory).

A.3.2 ogbg-molpcba

GCN: perturbation step size α=8e-03 for both the vanilla model and the one augmented by virtual node.

GIN: perturbation step size α=8e-03 for both the vanilla model and the one augmented by virtual node.

DeeperGCN: perturbation step size α=8e-03 with virtual node added, and the model is trained on NVIDIA Tesla V100 (32GB GPU memory).

A.3.3 ogbg-ppa

GCN: perturbation step size α=2e-03, when virtual node is added we use a larger α=5e-03.

GIN: perturbation step size α=8e-03, when virtual node is added we use a smaller α=5e-03.

DeeperGCN: perturbation step size α=8e-03, and the model is trained on NVIDIA Tesla V100 (32GB GPU memory).

A.3.4 ogbg-code

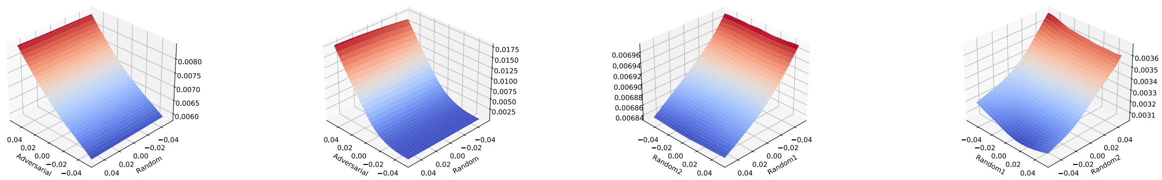
GCN: perturbation step size α=8e-03 for both the vanilla model and the one augmented by virtual node.

GIN: perturbation step size α=8e-03 for both the vanilla model and the one augmented by virtual node.

DeeperGCN: perturbation step size α=8e-03 with virtual node added, and the model is trained on NVIDIA Tesla V100 (32GB GPU memory).

B. Loss Landscape Visualization

Figure 6 shows the loss landscape of GIN model. We can see that our method further regularizes the loss landscape.



(a) FLAG adv-random

(b) Vanilla adv-random

(c) FLAG random-random

(d) Vanilla random-random

Figure 6. Loss landscape visualization. The test is conducted on one random validation graph from `ogbg-molhiv`. Two models are GIN trained with FLAG and a vanilla GIN. (a) and (b) projects loss onto a random direction and the other adversarial direction, while (c) and (d) use two random directions.

Name	#Nodes	#Edges	#Tasks	Train/Val/Test	Task Type	Metric
ogbn-products	2,449,029	61,859,140	1	8/2/90	Multi-class classification	Accuracy
ogbn-proteins	132,534	39,561,252	112	65/16/19	Binary classification	ROC-AUC
ogbn-arxiv	169,343	1,166,243	1	54/18/28	Multi-class classification	Accuracy
ogbn-mag	1,939,743	21,111,007	1	85/9/6	Multi-class classification	Accuracy
Cora	2485	5069	1	no official split$*$	Multi-class classification	Accuracy

Table 9. Node classification datasets statistics. $*$ denotes we follow the split of [21].

Name	#Nodes	#Edges	Train/Val/Test	Task Type	Metric
ogbl-ddi	4,267	1,334,889	80/10/10	Link prediction	Hits@20
ogbl-collab	235,868	1,285,465	92/4/4	Link prediction	Hits@50

Table 10. Link prediction datasets statistics.

Name	#Graphs	Avg #Nodes	Avg #Edges	#Tasks	Train/Val/Test	Task Type	Metric
ogbg-molhiv	41,127	25.5	27.5	1	80/10/10	Binary classification	ROC-AUC
ogbg-molpcba	437,929	26.0	28.1	128	80/10/10	Binary classification	AP
ogbg-ppa	158,100	243.4	2,266.1	1	49/29/22	Multi-class classification	Accuracy
ogbg-code	452,741	125.2	124.2	1	90/5/5	Sub-token prediction	F1 score

Table 11. Graph classification datasets statistics.