Table 3. Dataset statistics.

Dataset	#Nodes	#Edges	#Features	#Classes
Cora	2,485	5,429	1,433	7
Citeseer	2,110	3,757	3,703	6
BlogCataLog	5,196	343,486	8,189	6

Algorithm 1 Certified robustness inspired PGD (CR-PGD) graph evasion attack to GNNs

Input: Node classifier f, graph $G(\mathbf{A})$, testing nodes $\mathcal{V}_{Te}$, perturbation budget Δ, total iterations T, #samples N, noise parameter β, confidence level $1-\alpha$, a, interval INT.
Output: Adversarial graph perturbation $\delta^{(T)}$.
Initialize: $t=0$; graph perturbation $\delta^{(0)}=0$;
while $t<T$ **do**
  // Stage 1: Obtaining the CR inspired loss
  if *t mod INT != 0* **then**
    Reuse the node weights: $w^{(t)}(v)=w^{(t-1)}(v)$;
  else
    Define the perturbed graph: $\mathbf{A}^{(t)}=\mathbf{A}\oplus\delta^{(t)}$;
    Sample N noise matrices $\{\epsilon^j\}_{j=1}^N$ from the noise distribution Equation 7 with parameter β;
    for *each node* $v\in\mathcal{V}_{Te}$ **do**
      Compute the frequency N_{y_v} for label y_v: $N_{y_v}=\sum_{j=1}^N \mathbb{I}(f(\mathbf{A}^{(t)}\oplus\epsilon^j;v)=y_v)$;
      Estimate the low bound probability $\underline{p_{y_v}}$ with confidence $1-\alpha$: $\underline{p_{y_v}}=B(\alpha;N_{y_v},N-N_{y_v}+1)$;
      Calculate the certified perturbation size $K(\underline{p_{y_v}})$ using $\underline{p_{y_v}}$ and algorithm in [35];
      Assign a weight $w(v)$ to each node v: $w^{(t)}(v)=\frac{1}{1+\exp(a\cdot K(\underline{p_{y_v}}))}$;
    end
  end
  Define the certified robustness inspired test loss: $\mathcal{L}_{CR}(f,\mathbf{A}^{(t)},\mathcal{V}_{Te})=\sum_{v\in\mathcal{V}_{Te}} w^{(t)}(v)\ell(f(\mathbf{A}^{(t)};v),y_v)$;
  // Stage 2: Running the PGD attack with CR loss
  $\delta^{(t+1)}=\text{Proj}_{\mathbb{B}}(\delta^{(t)}+\eta\cdot\nabla_{\delta^{(t)}}\mathcal{L}_{CR}(f,\mathbf{A}^{(t)},\mathcal{V}_{Te})$;
  Update $t=t+1$.
end
return $\delta^{(T)}$

Algorithm 2 Certified robustness inspired Minmax (CR-Minmax) graph poisoning attack to GNNs

Input: GNN algorithm $\mathcal{A}$, Graph $G(\mathbf{A})$, training nodes $\mathcal{V}_{Tr}$, perturbation budget Δ, number of samples N, noise parameter β, confidence level $1-\alpha$, a, interval INT.
Output: Adversarial graph perturbation $\delta^{(T)}$.
Initialize: $t=0$; $\delta^{(0)}=0$; random/pretrained GNN model $\theta^{(0)}$;
while $t<T$ **do**
  // Stage 1: Obtaining the CR inspired loss
  if *t mod INT != 0* **then**
    Reuse the node weights: $w^{(t)}(v)=w^{(t-1)}(v)$;
  else
    Define the perturbed graph: $\mathbf{A}^{(t)}=\mathbf{A}\oplus\delta^{(t)}$;
    Sample N noise matrices $\{\epsilon^j\}_{j=1}^N$ from the noise distribution Equation 7 with parameter β;
    Train N node classifiers $\{\tilde{f}^n\}$ with current perturbed graph $\mathbf{A}^{(t)}$ with the N sampled noisy matrices $\{\epsilon^n\}$: $\tilde{f}^1=\mathcal{A}(\mathbf{A}^{(t)}\oplus\epsilon^1,\mathcal{V}_{Tr}),\cdots,\tilde{f}^N=\mathcal{A}(\mathbf{A}^{(t)}\oplus\epsilon^N,\mathcal{V}_{Tr})$
    for *each node* $v\in\mathcal{V}_{Tr}$ **do**
      Compute the frequency N_{y_v} for label y_v: $N_{y_v}=\sum_{j=1}^N \mathbb{I}(\tilde{f}^j(\mathbf{A}^{(t)}\oplus\epsilon^j;v)=y_v)$;
      Estimate the low bound probability $\underline{p_{y_v}}$ with confidence $1-\alpha$: $\underline{p_{y_v}}=B(\alpha;N_{y_v},N-N_{y_v}+1)$;
      Calculate the certified perturbation size $K(\underline{p_{y_v}})$ using $\underline{p_{y_v}}$ and algorithm in [35];
      Assign a weight $w(v)$ to each node v: $w^{(t)}(v)=\frac{1}{1+\exp(a\cdot K(\underline{p_{y_v}}))}$;
    end
  end
  Define the certified robustness inspired training loss: $\mathcal{L}_{CR}(f,\mathbf{A}^{(t)},\mathcal{V}_{Tr})=\sum_{v\in\mathcal{V}_{Tr}} w_v^{(t)}\cdot\ell(f(\mathbf{A}^{(t)};v),y_v)$;
  // Stage 2: Running the Minmax attack with CR loss
  Step 1: Inner minimization over model parameter θ: $\theta^{(t+1)}=\theta^{(t)}-\eta_1\nabla_\theta\mathcal{L}_{CR}(f_{\theta^{(t)}},\mathbf{A}^{(t)},\mathcal{V}_{Tr})$;
  Step 2: Outer maximization over graph perturbation δ: $\delta^{(t+1)}=\text{Proj}_{\mathbb{B}}(\delta^{(t)}+\eta_2\nabla_\delta\mathcal{L}_{CR}(f_{\theta^{(t+1)}},\mathbf{A}^{(t)},\mathcal{V}_{Tr}))$;
  Update $t=t+1$.
end
return $\delta^{(T)}$