Local Graph Clustering with Noisy Labels

Artur Back de Luca, Kimon Fountoulakis, and Shenghao Yang

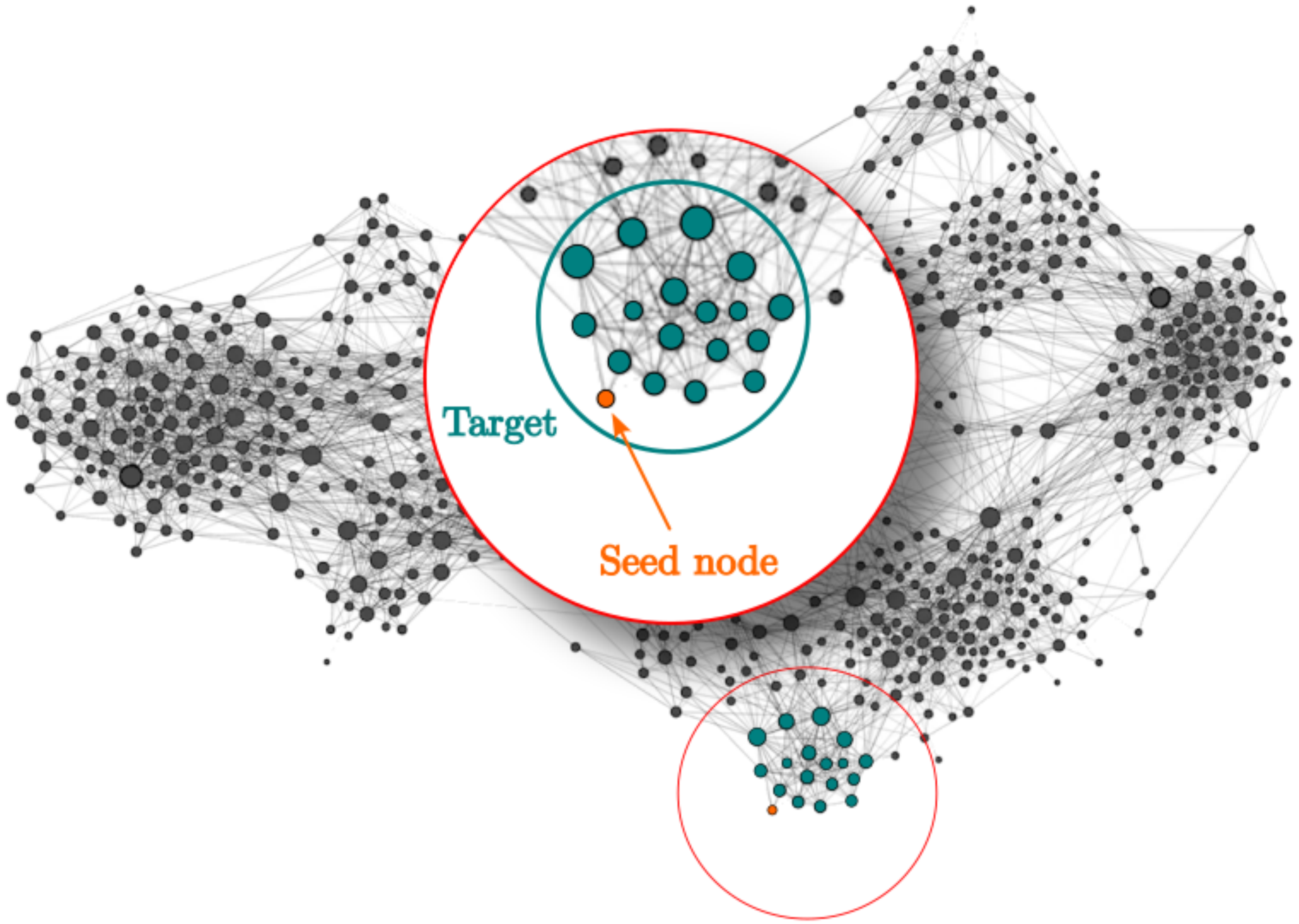
UNIVERSITY OF WATERLOO | DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

Local graph clustering

Setting: Given a graph $G = (V, E)$, and a seed node $s \in V$

Goal: Find a good cluster that contains s, without necessarily exploring the whole graph

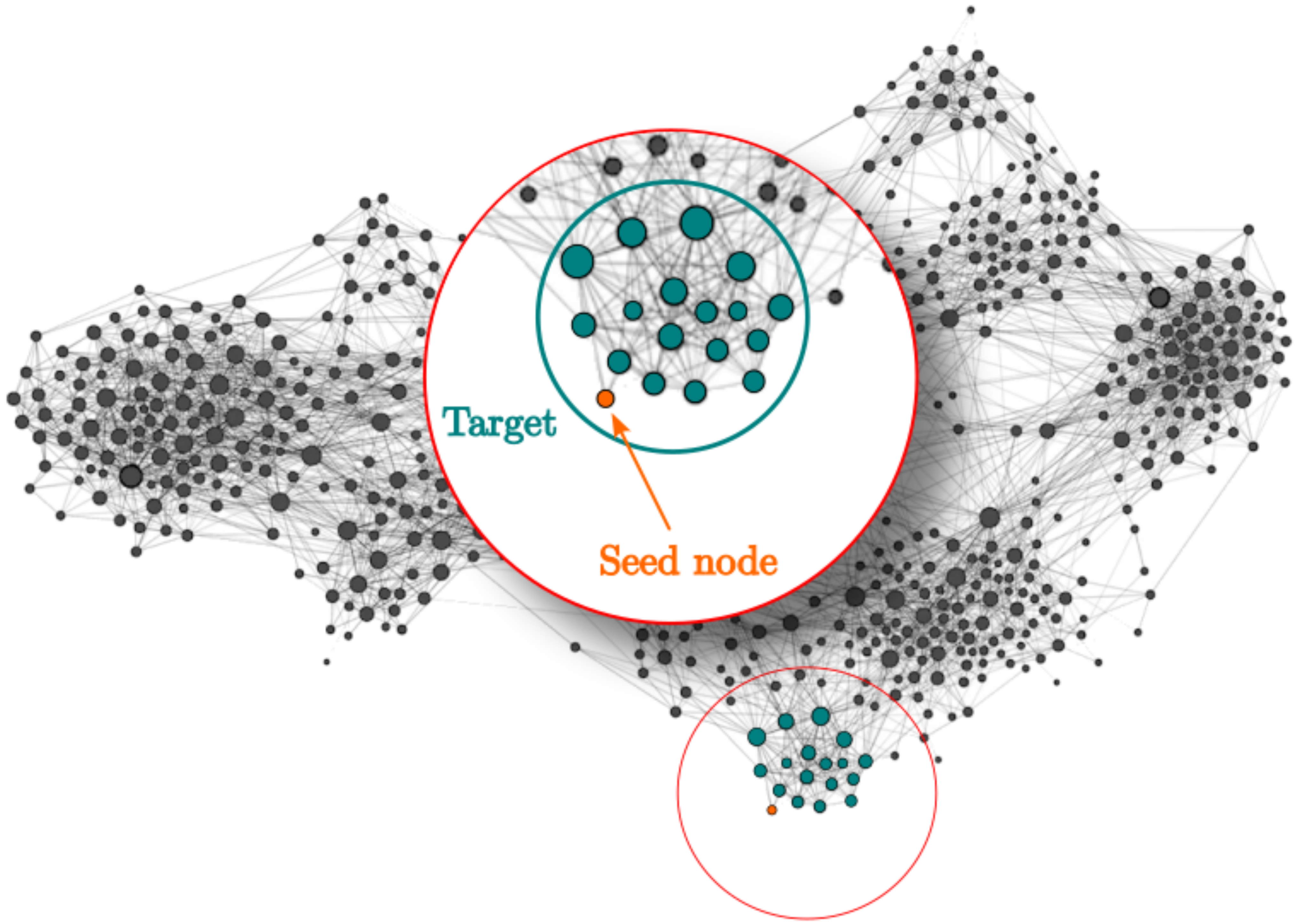
Random walk [Spielman & Teng 2013]
PageRank [ACL 2006]
Heat kernel [Chung 2007]
Evolving sets [Andersen & Peres 2008]
Capacity releasing diffusion [Di *et al* 2017]
Flow diffusion [Fountoulakis *et al* 2020]
and many more…



Local graph clustering

Setting (this work): Given a graph $G = (V, E)$ **with noisy node labels**, and a seed node $s \in V$

Goal: Find a good cluster that contains s, without necessarily exploring the whole graph



Contributions

- A simple algorithm integrates noisy node labels into local graph clustering, demonstrating their usefulness, particularly when the graph structure is poor.
- We provide a theoretical analysis on the recovery of an unknown target cluster in a local random graph model with additional noisy node labels
- We empirically verify the results through extensive experiments over both synthetic and real-world data

Noisy node labels

- Each node receives a binary label indicating its membership: 1 if it belongs to the target cluster and 0 if it does not. A fraction of the labels is then flipped to introduce label noise

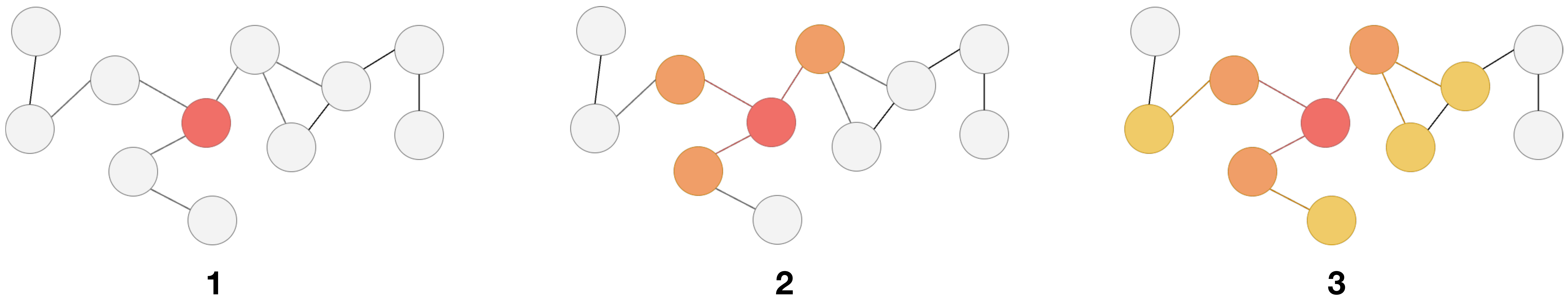
- From a practical point of view, noisy labels can be the result of an imperfect classifier that predicts cluster affiliation based on node attributes
    - This allows us work with text, image, audio, etc.

- By abstracting all sources of information as noisy labels, we can theoretically study the benefit of incorporating additional information without explicit assumptions on node attributes

Local graph diffusion

- Generic process to spread mass from a seed node to nearby nodes via edges in the graph

- Mass tends to spread within well-connected clusters



Local graph clustering

- **Input:** Graph $G = (V, E)$, seed node $s \in V$
- **Algorithm** (informal)**:**
  - Run local graph diffusion in G starting from s
  - Check where and how the mass spread within G around s
  - Obtain an output cluster (by applying rounding/post-precessing)

Local graph clustering with noisy labels

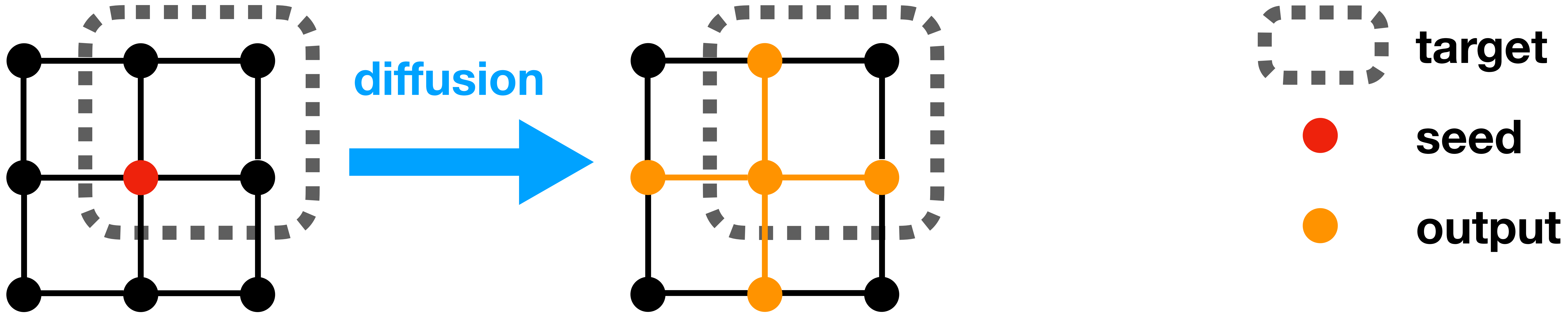
- **Input:** Graph $G = (V, E)$, seed node $s \in V$, noisy node labels $\tilde{y}_i \in \{0,1\}$, $\forall i$
- **Algorithm** (informal)**:**
  - Define weighted graph $G' = (V, E, w)$ with edge weight

$$w_{ij} = \begin{cases} 1 & \text{if } \tilde{y}_i = \tilde{y}_j, \\ \varepsilon & \text{if } \tilde{y}_i \neq \tilde{y}_j, \quad \varepsilon \in [0,1) \end{cases}$$

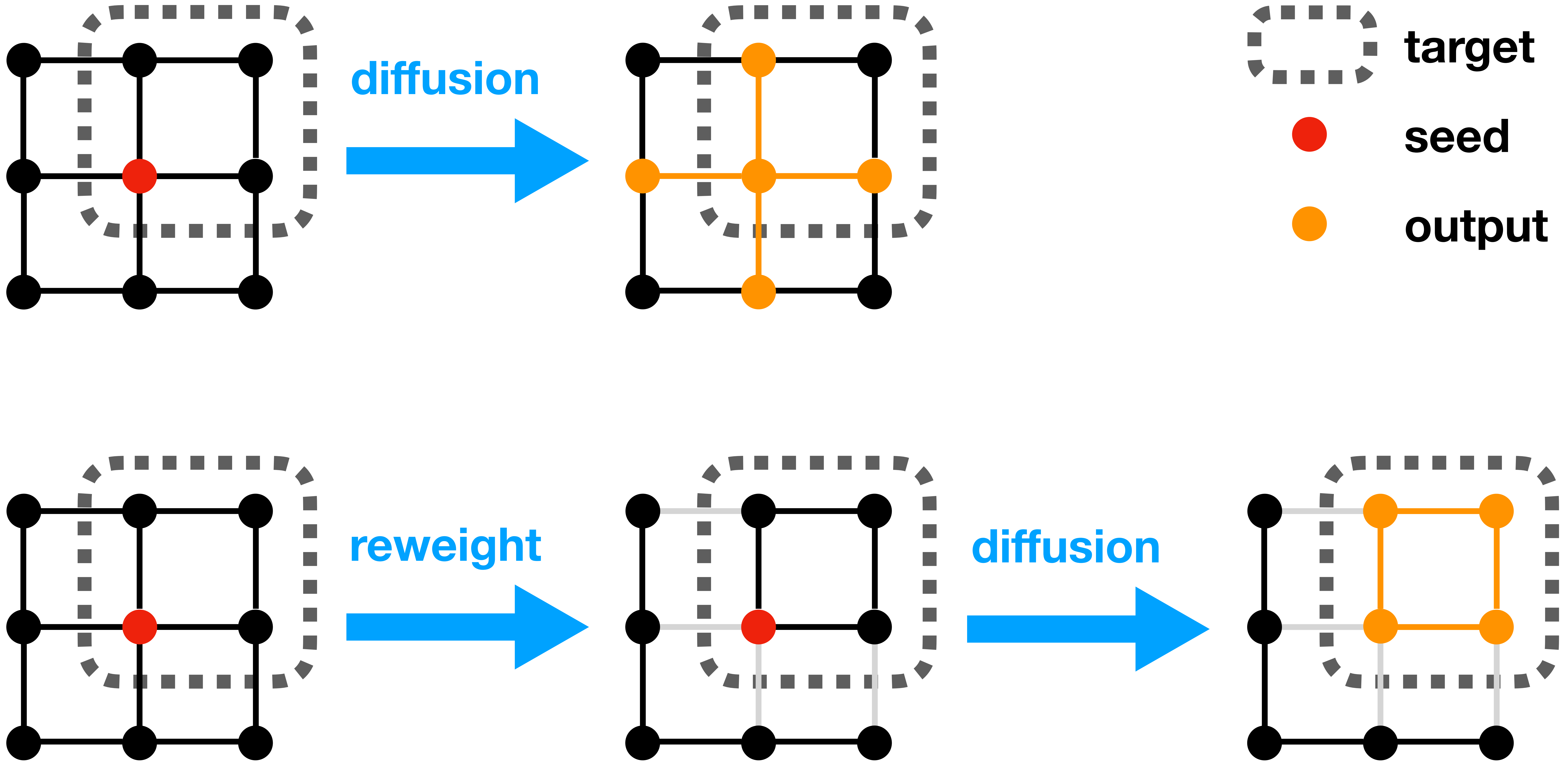
  - Run weighted local graph diffusion in G' starting from s
  - Check where and how the mass spread within G' around s
  - Obtain an output cluster (by applying rounding/post-precessing)

How does reweighing edges help exactly?

Example: how edge weights can help



Example: how edge weights can help



Local random model with noisy labels

Local random graph: Given a set of nodes V and a target cluster $K \subset V$

- Draw an edge (i, j) with probability p if $i \in K, j \in K$
- Draw an edge (i, j) with probability q if $i \in K, j \notin K$
- Edges (i, j) where $i, j \notin K$ can be arbitrary
- **Structural signal** $\gamma = \big(p\,(|K| - 1)\big) \big/ \big(q\,(n - |K|)\big)$

Noisy labels: Every node $i \in V$ is assigned a binary label $\tilde{y}_i \in \{0,1\}$

- $\tilde{Y}_1 = \{i \in V : \tilde{y}_i = 1\}$ and $\tilde{Y}_0 = \{i \in V : \tilde{y}_i = 0\}$
- **Label accuracy** $a_1 = |\tilde{Y}_1 \cap K| \big/ |K|$ and $a_0 = |\tilde{Y}_0 \cap K^C| \big/ |K^C|$

Recovery guarantees

- Suppose that $p = \omega(\sqrt{\log |K|} / \sqrt{|K|})$

- Let S^* be the output of diffusion in the **weighted graph**, then

$$\mathrm{F1}(S^*) \gtrsim \left[1 + \frac{(1-a_1)}{2} + \frac{(1-a_0)}{2\gamma} + \frac{(1-a_0)^2}{2a_1\gamma^2}\right]^{-1}$$

- **For comparison:** Let $S^\dagger$ be the output of diffusion in the **original graph**, then

$$\mathrm{F1}(S^\dagger) \gtrsim \left[1 + \frac{1}{\gamma} + \frac{1}{2\gamma^2}\right]^{-1}$$

Comparison with SOTA on real data

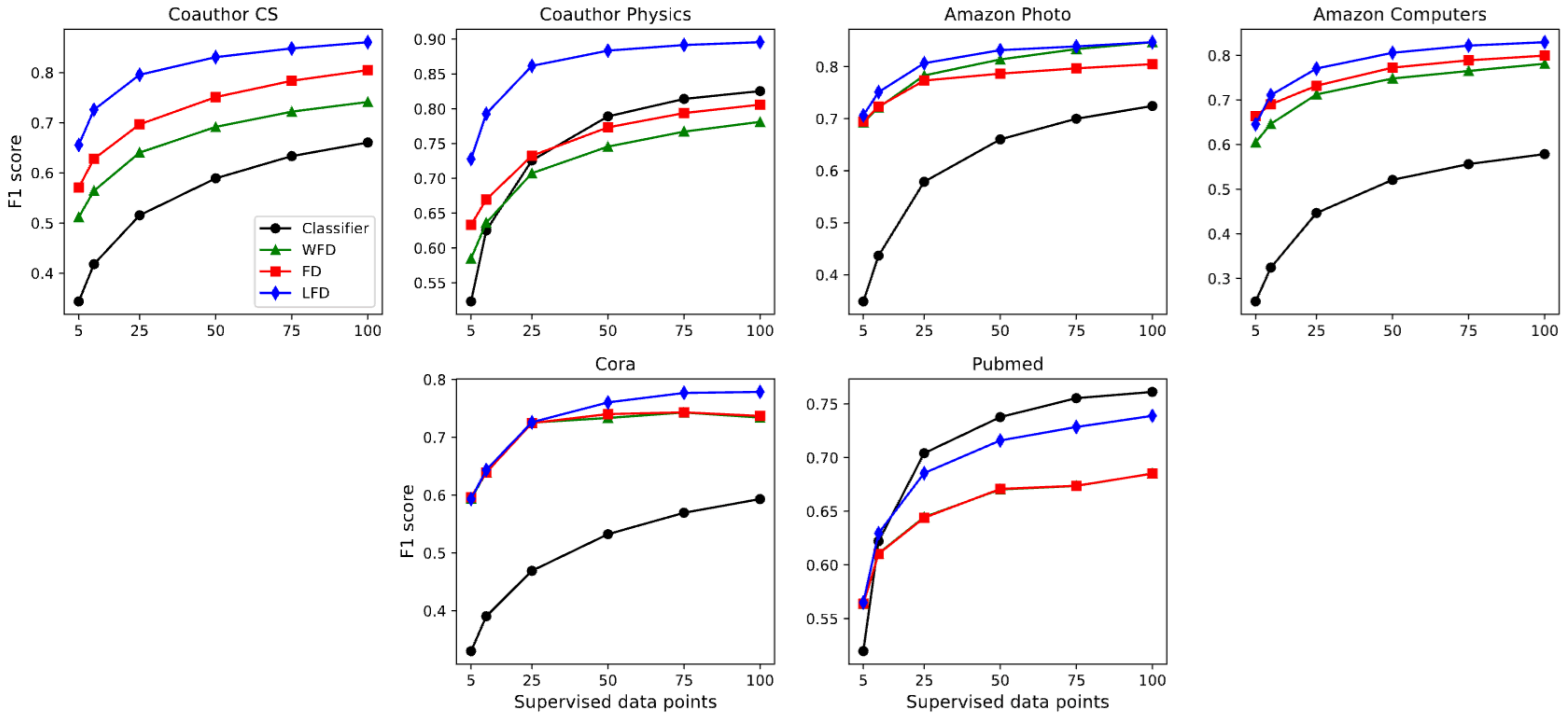


Figure 3: F1 scores for local clustering using Flow Diffusion (FD), Weighted Flow Diffusion (WFD), Label-based Flow Diffusion (LFD), and Logistic Regression (Classifier) with an increasing number of positive and negative ground-truth samples.

Improvement as high as 13% over any other method