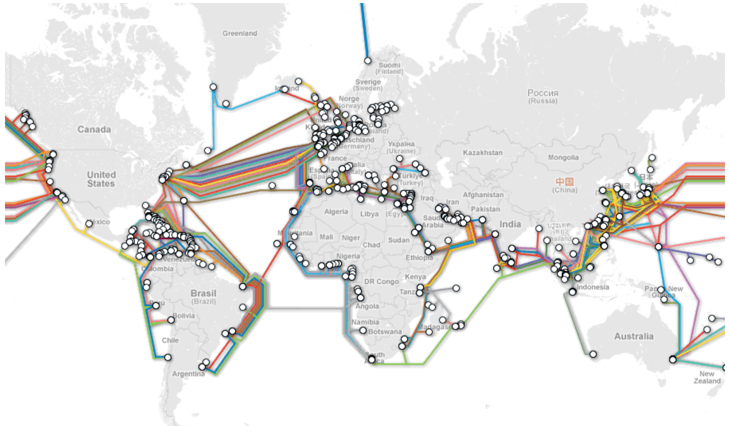
SFU

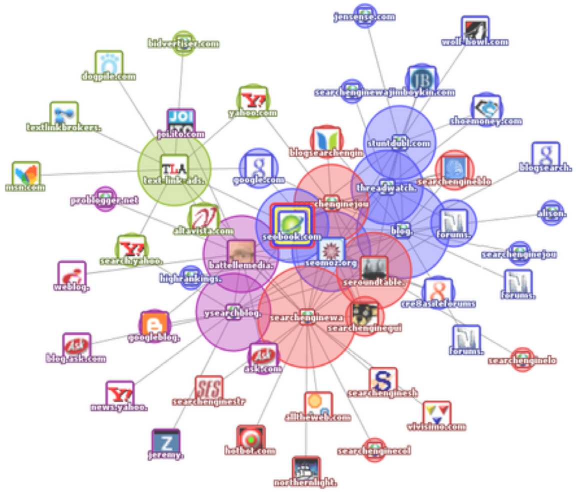
GraphBolt: Dependency-Driven Synchronous Processing of Streaming Graphs

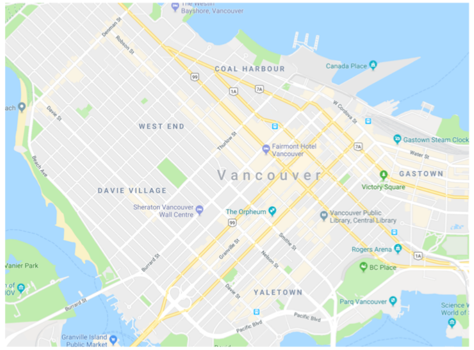
Mugilan Mariappan and Keval Vora
Simon Fraser University

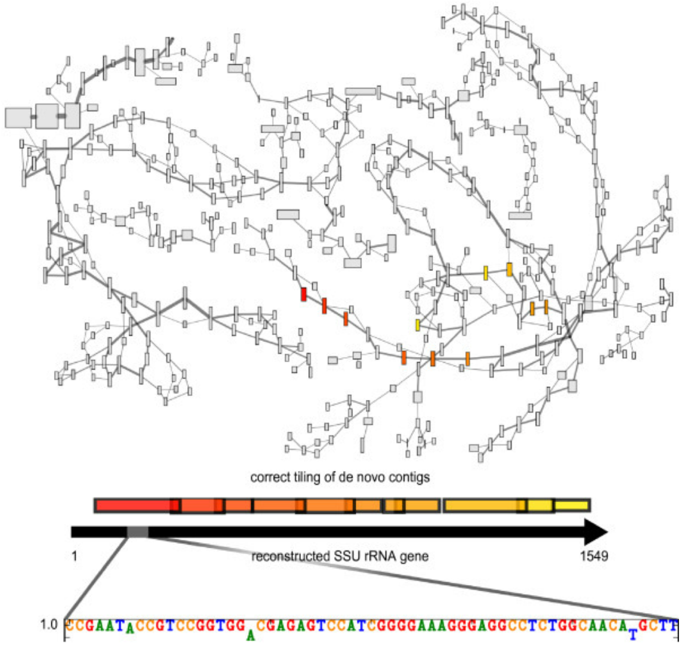
EuroSys'19 – Dresden, Germany
March 27, 2019

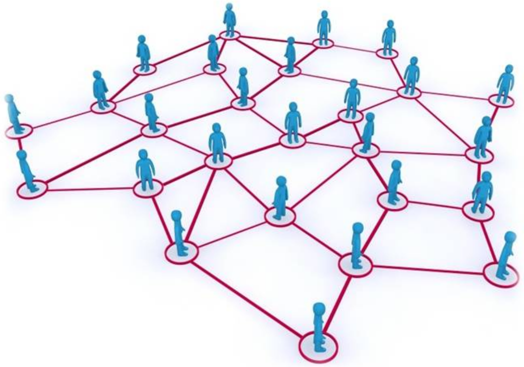
Graph Processing

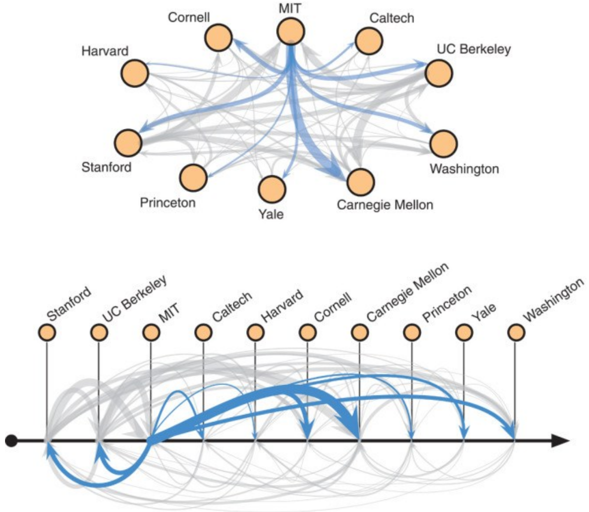












Dynamic Graph Processing

Real-time Processing

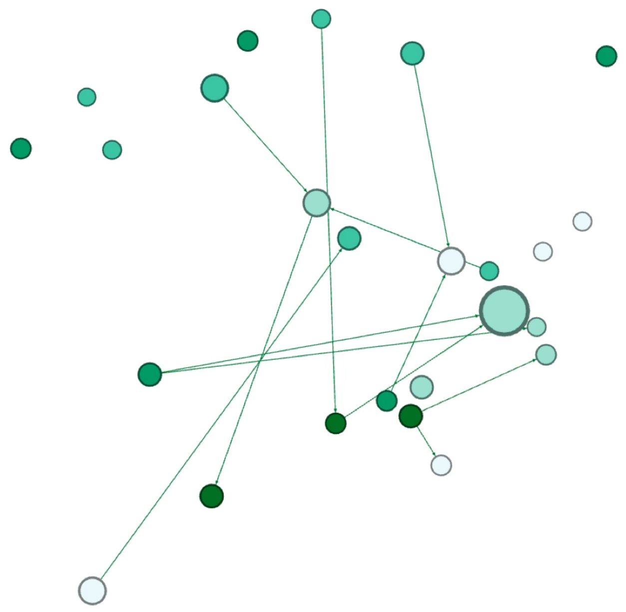
- Low Latency

Real-time **Batch** Processing

- High Throughput

THENEXTPLATFORM

Alipay payments unit of Chinese retailer Alibaba [..] has 120 billion nodes and over 1 trillion relationships [..]; this graph has **2 billion updates each day** and was running at **250,000 transactions per second** on Singles Day [..]



The Graph Database Poised to Pounce on The Mainstream. Timothy Prickett Morgan, The Next Platform. September 19, 2018.

Streaming Graph Processing

Incremental Processing

- Adjust results **incrementally**
- Reuse work that has already been done

Tornado [SIGMOD'16]

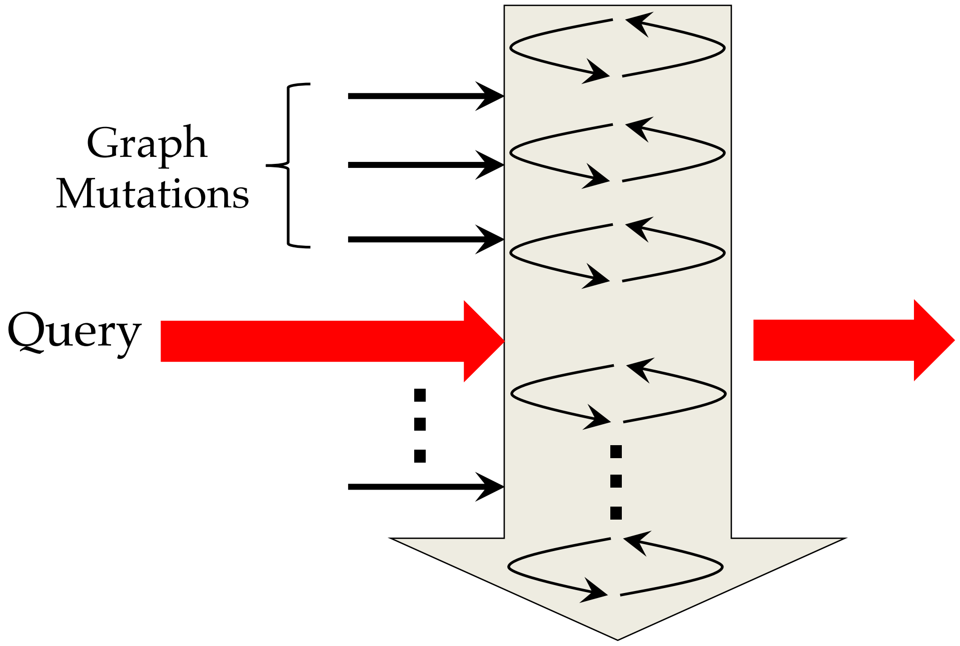
GraphIn [EuroPar'16]

KineoGraph [EuroSys'12]

Tag Propagation
upon mutation

Over 75% values get thrown out

KickStarter [ASPLOS'17]



KickStarter: Fast and Accurate Computations on Streaming Graph via Trimmed Approximations. Vora et al., ASPLOS 2017.

Streaming Graph Processing

Incremental Processing

- Adjust results **incrementally**
- Reuse work that has already been done

Tornado [SIGMOD'16]

GraphIn [EuroPar'16]

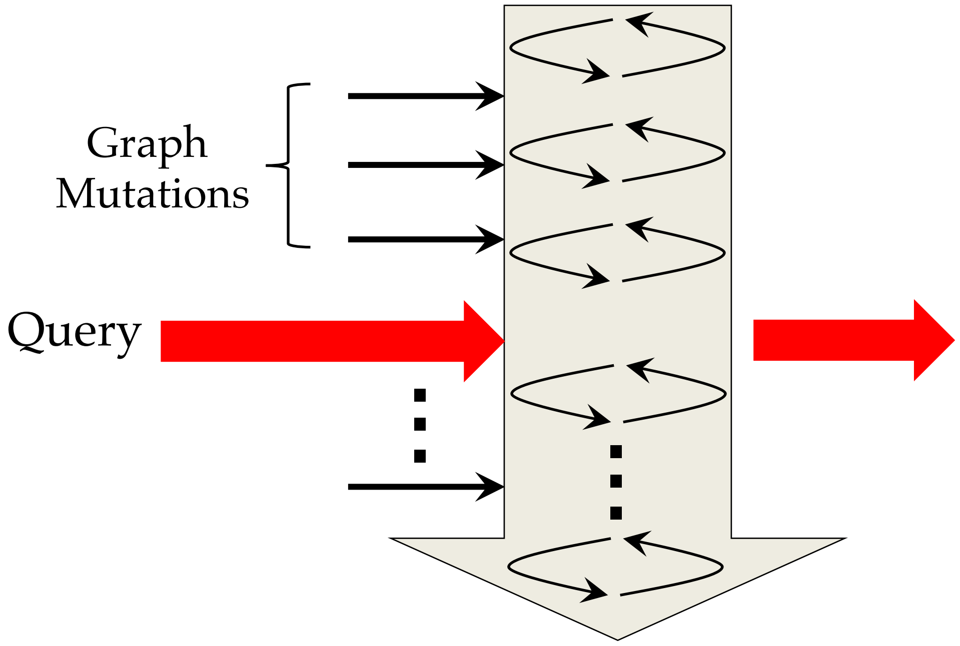
KineoGraph [EuroSys'12]

KickStarter [ASPLOS'17]

Maintain **Value Dependences**

Incrementally refine results

Less than 0.0005% values thrown out



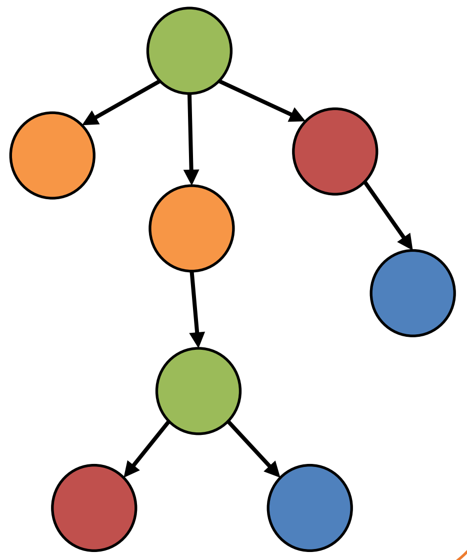
KickStarter: Fast and Accurate Computations on Streaming Graph via Trimmed Approximations. Vora et al., ASPLOS 2017.

Streaming Graph Processing

Incremental Processing

- Adjust results **incrementally**
- Reuse work that has already been done

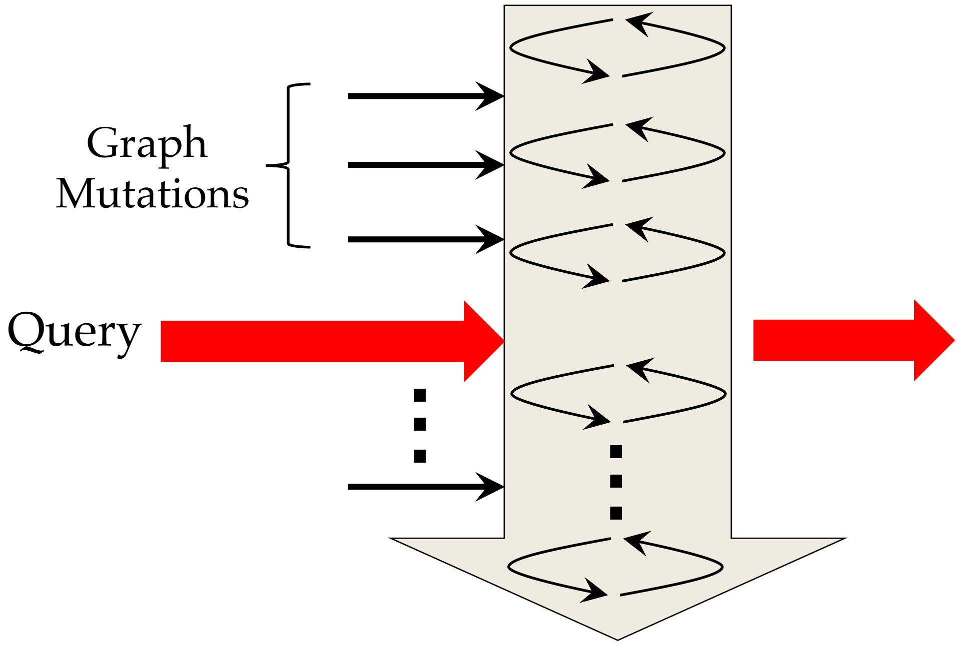
Monotonic Graph Algorithms



KickStarter [ASPLOS'17]

Maintain **Value Dependences**
Incrementally refine results

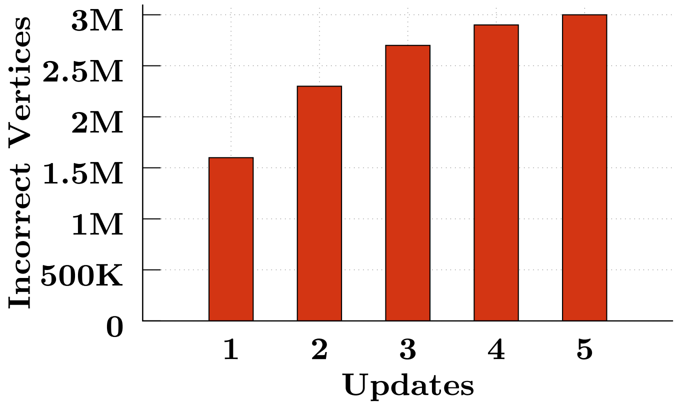
Less than 0.0005% values thrown out

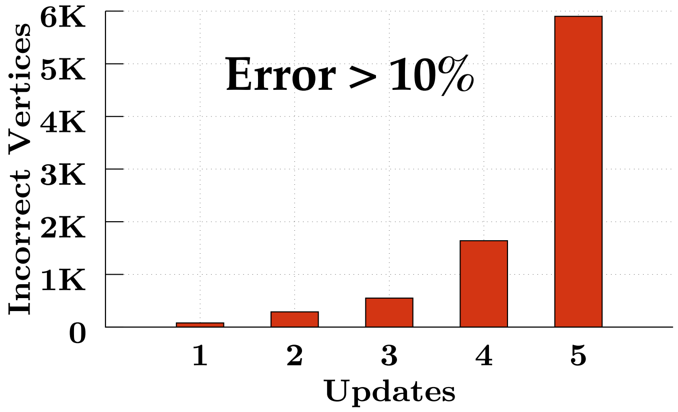


KickStarter: Fast and Accurate Computations on Streaming Graph via Trimmed Approximations. Vora et al., ASPLOS 2017.

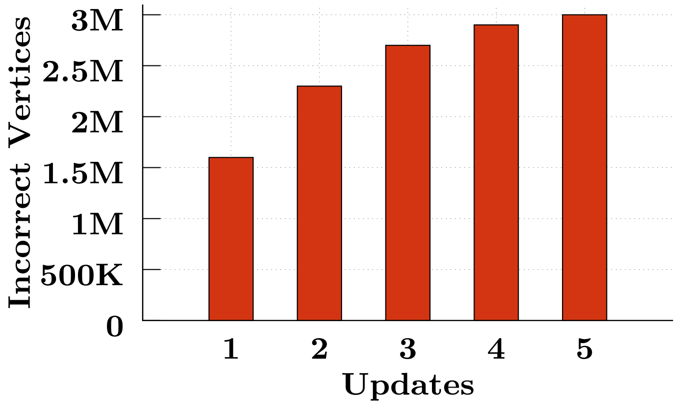
Streaming Graph Processing

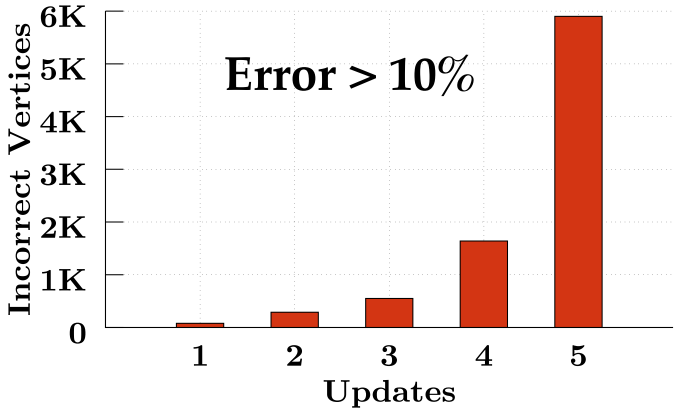
- Belief Propagation
- Co-Training Expectation Maximization
- Collaborative Filtering
- Label Propagation
- Triangle Counting
- …



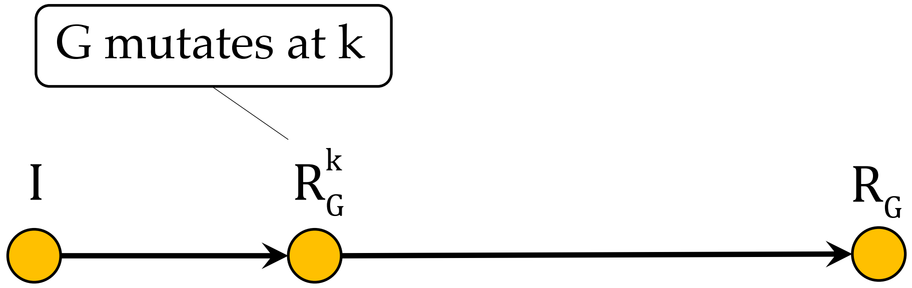


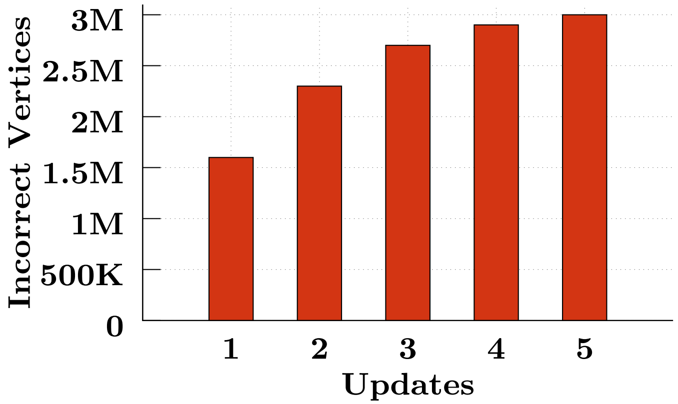
Streaming Graph Processing

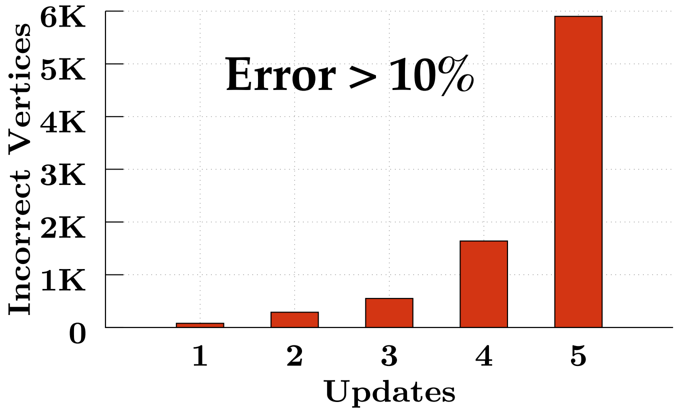




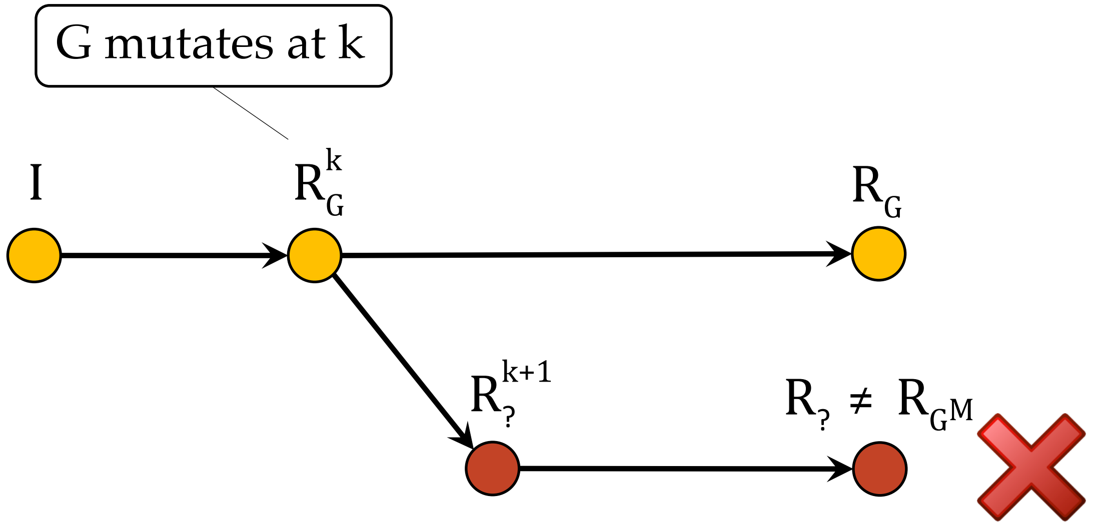
Streaming Graph Processing

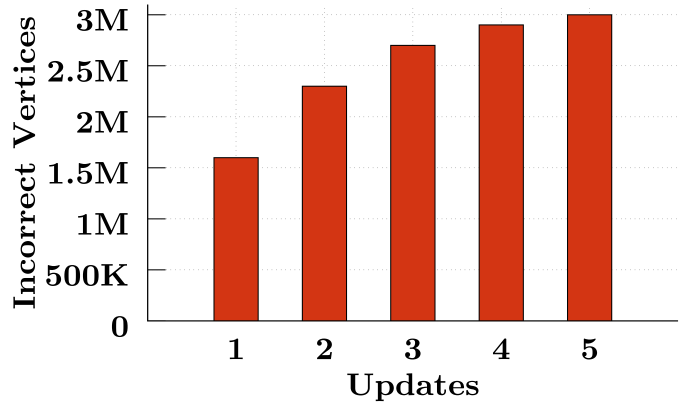


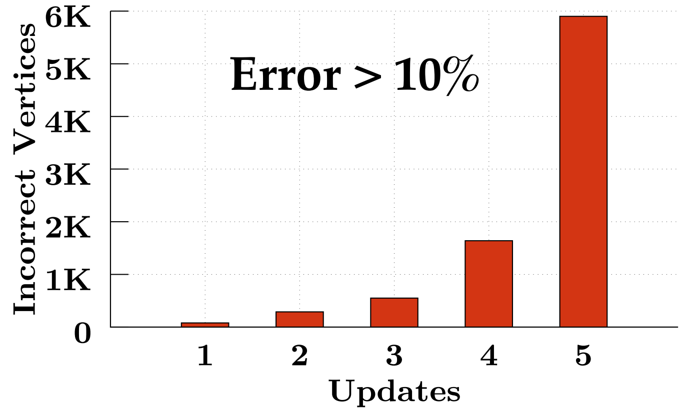




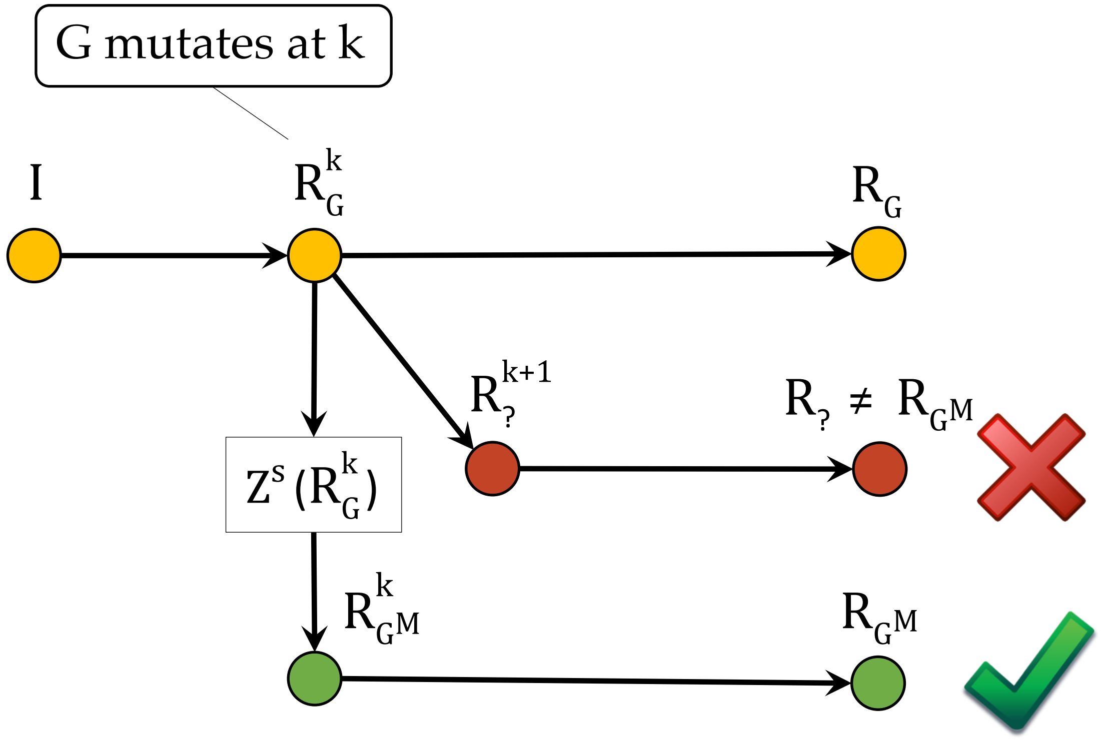
Streaming Graph Processing

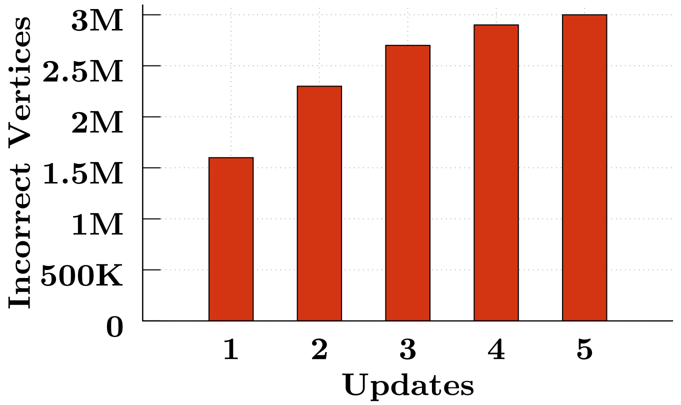


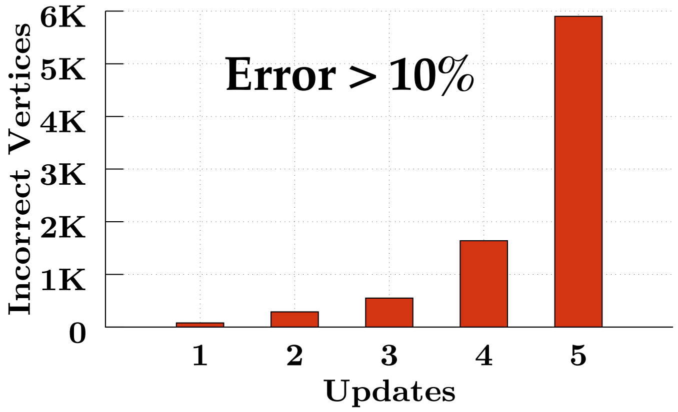




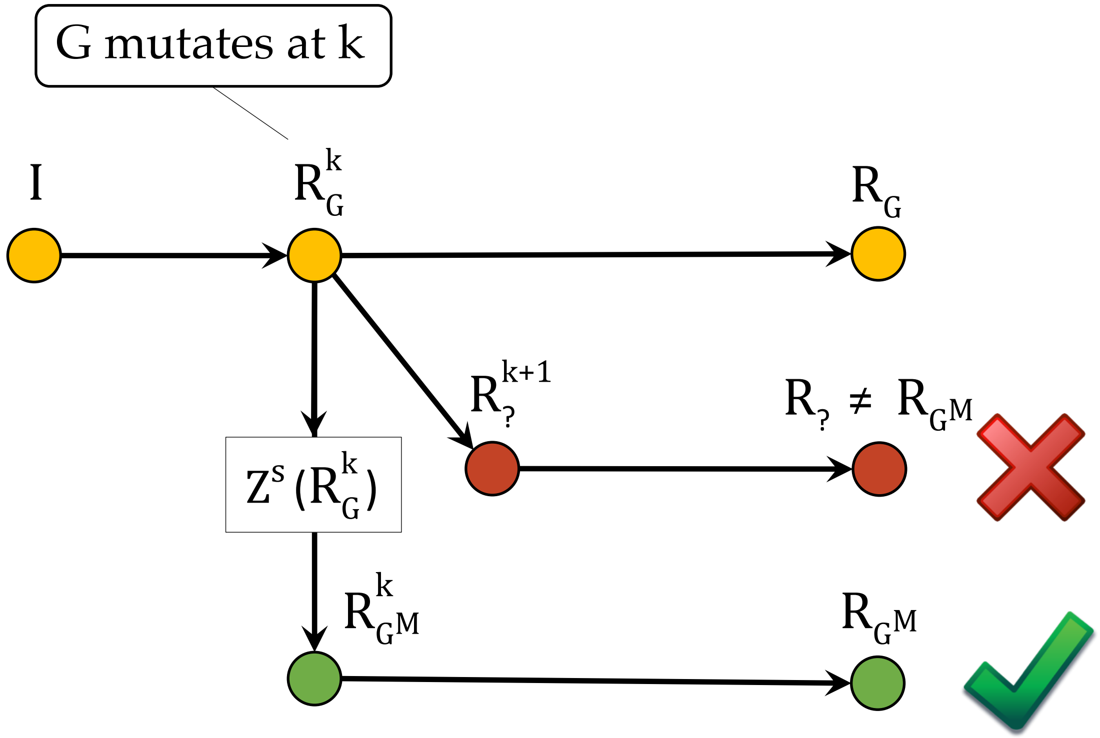
Streaming Graph Processing







GraphBolt

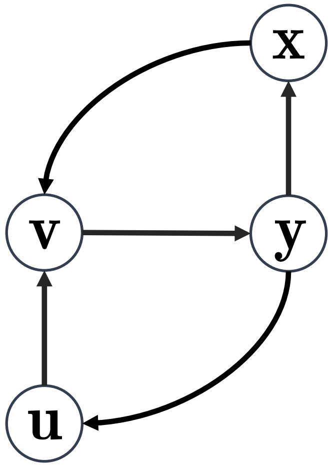


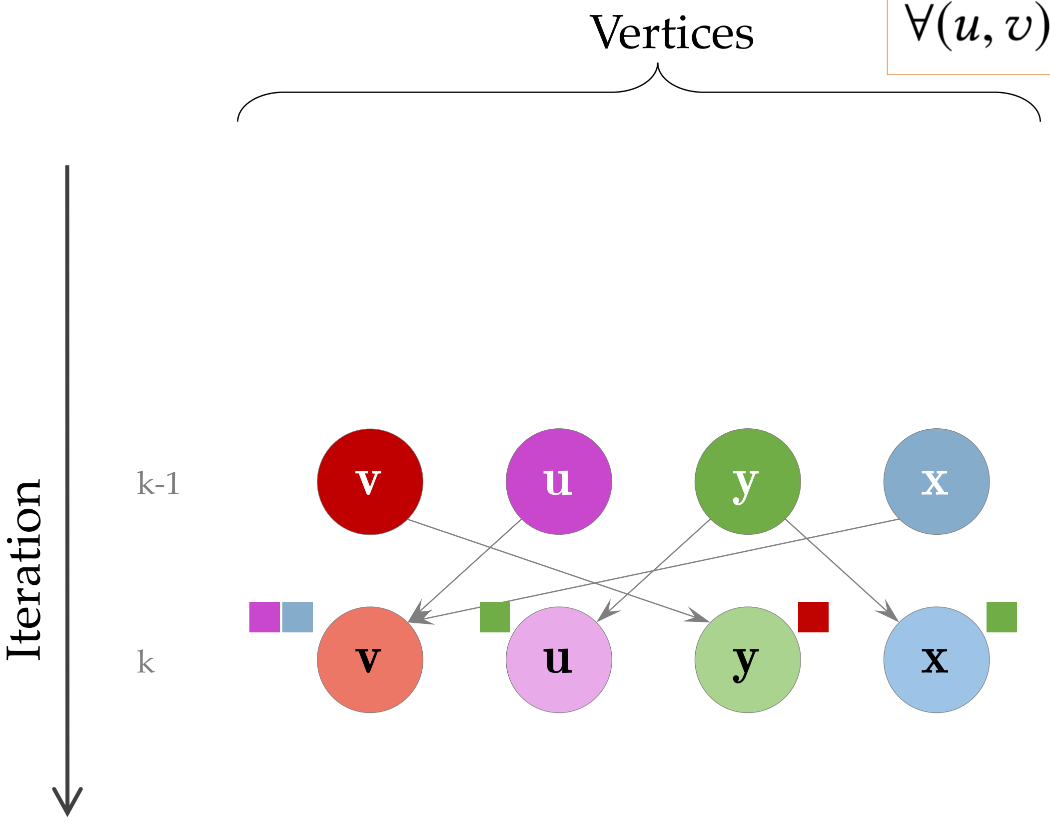
- **Dependency-Driven Incremental** Processing of Streaming Graphs
- Guarantee **Bulk Synchronous Parallel** Semantics

- Lightweight dependence tracking
- Dependency-aware value refinement upon graph mutation

Bulk Synchronous Processing (BSP)

$$\boxed{\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}}$$

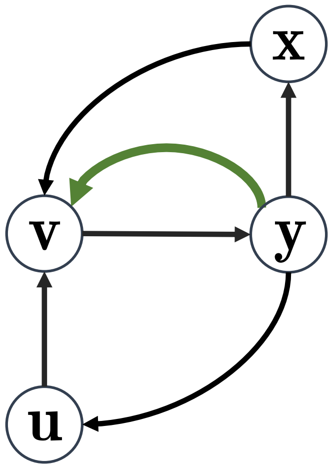




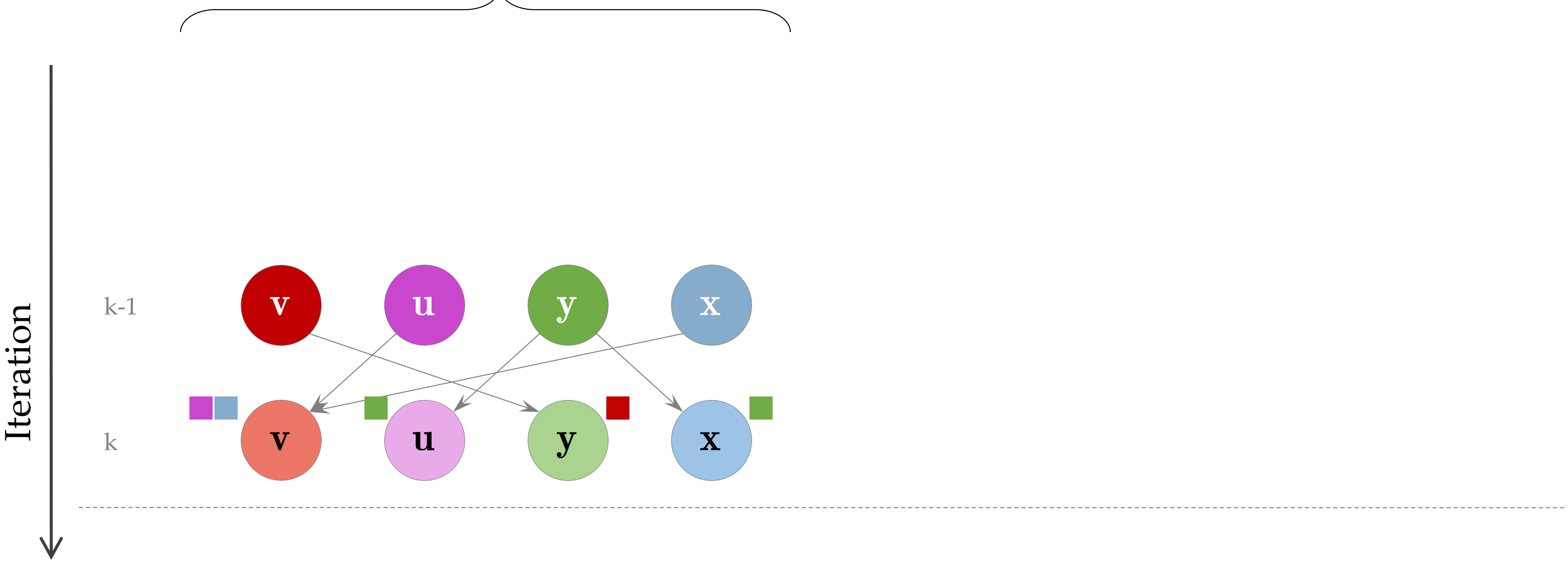
Streaming Graph

Upon Edge Addition

$$\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}$$



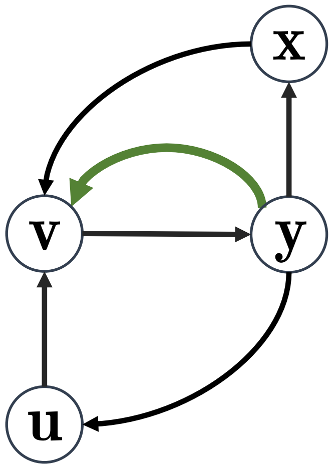
Vertices

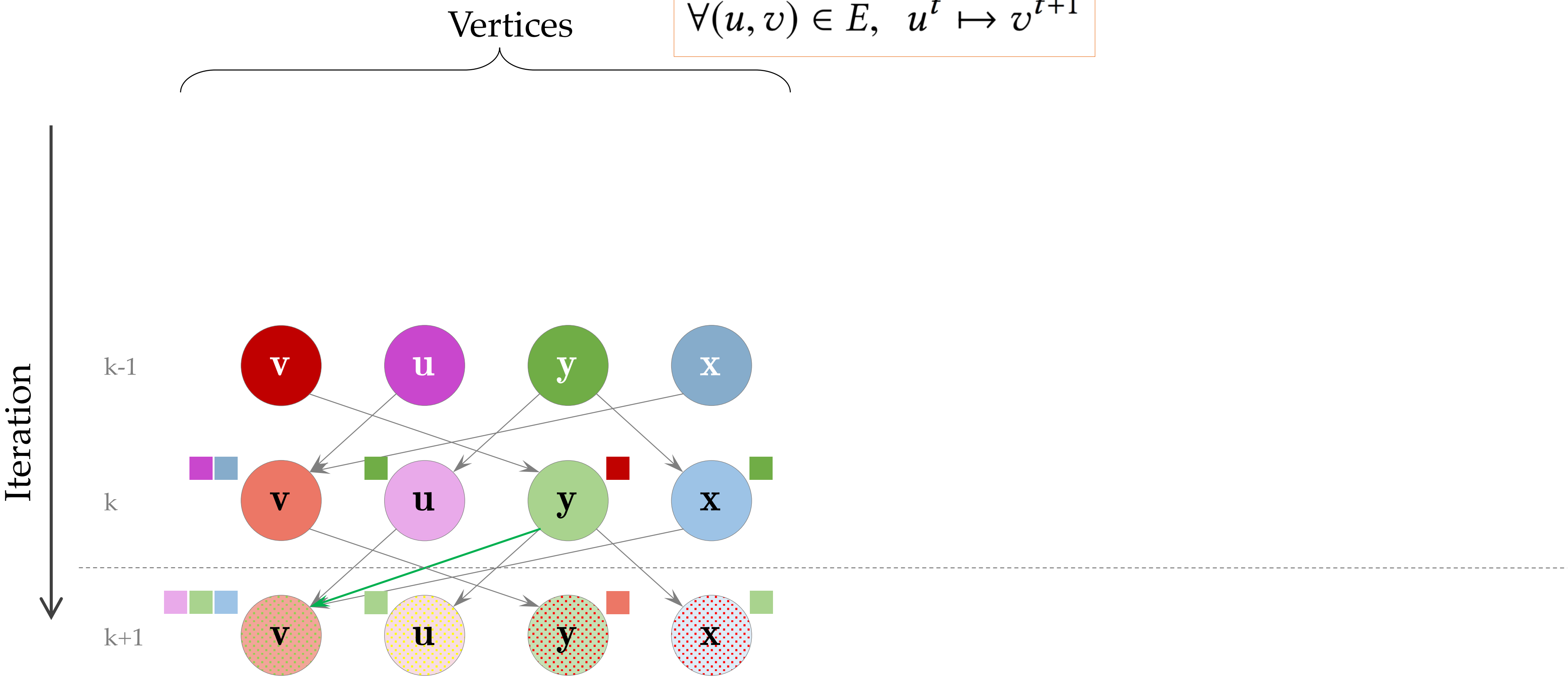


Streaming Graph

Upon Edge Addition

$$\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}$$

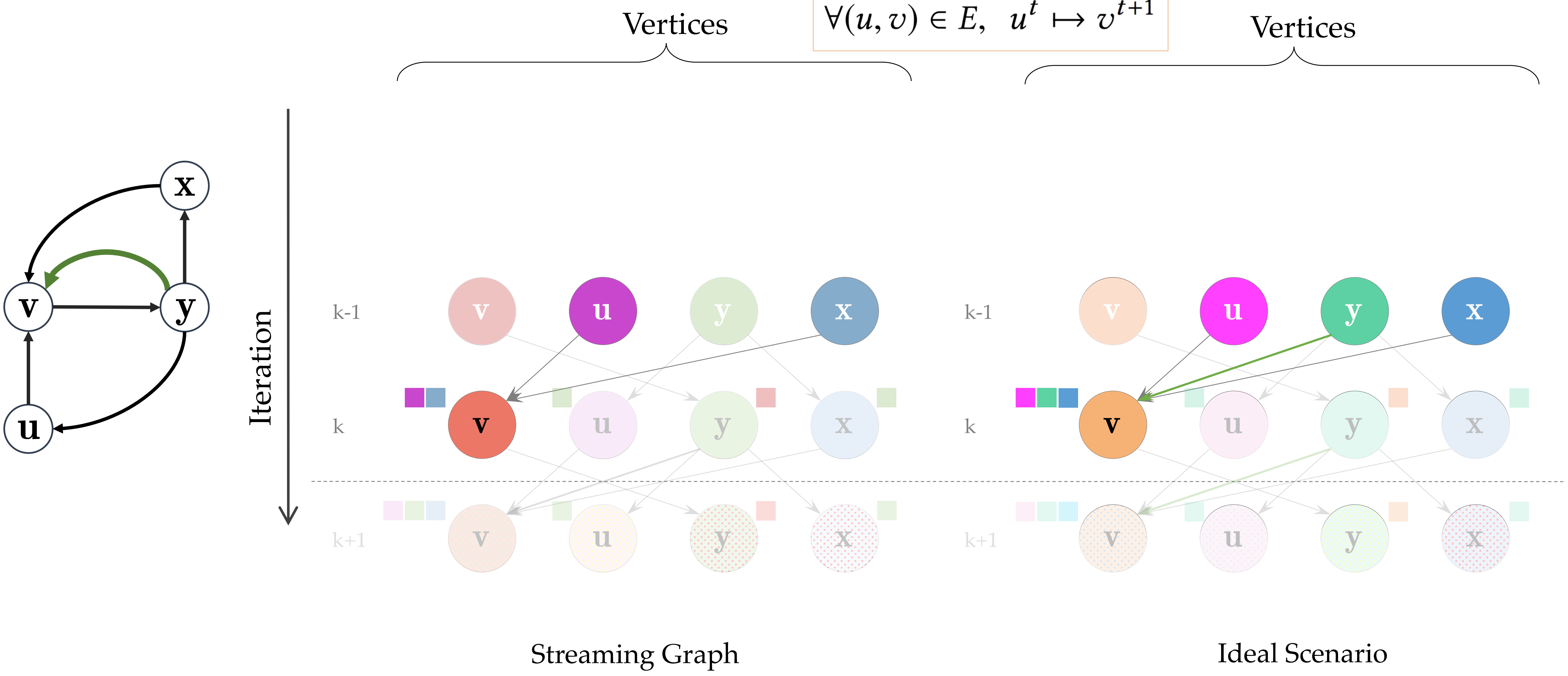




Streaming Graph

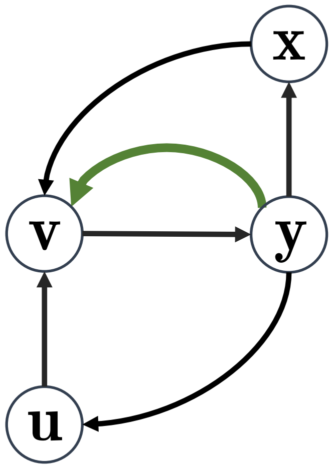
Upon Edge Addition

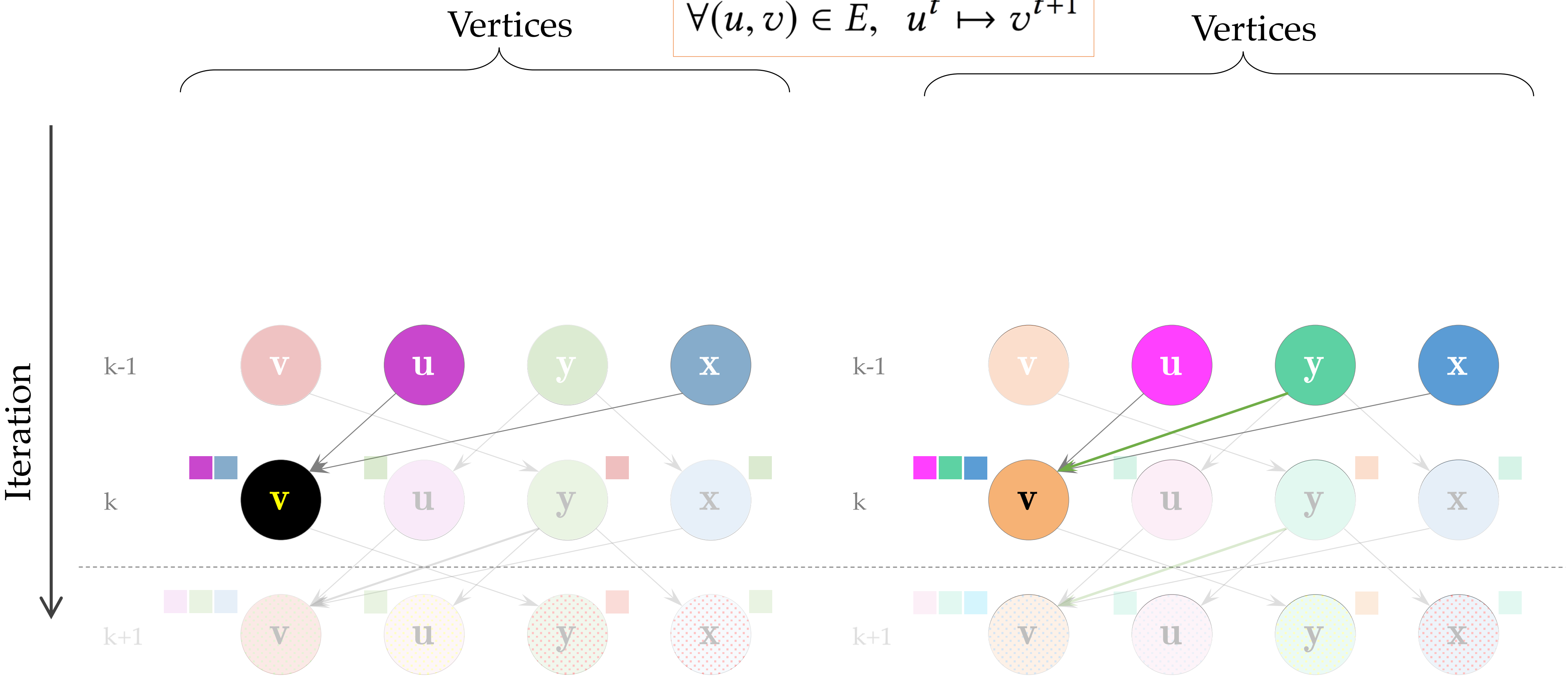
$$\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}$$



Upon Edge Addition

$$\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}$$



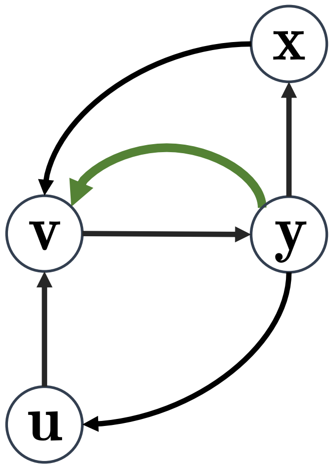


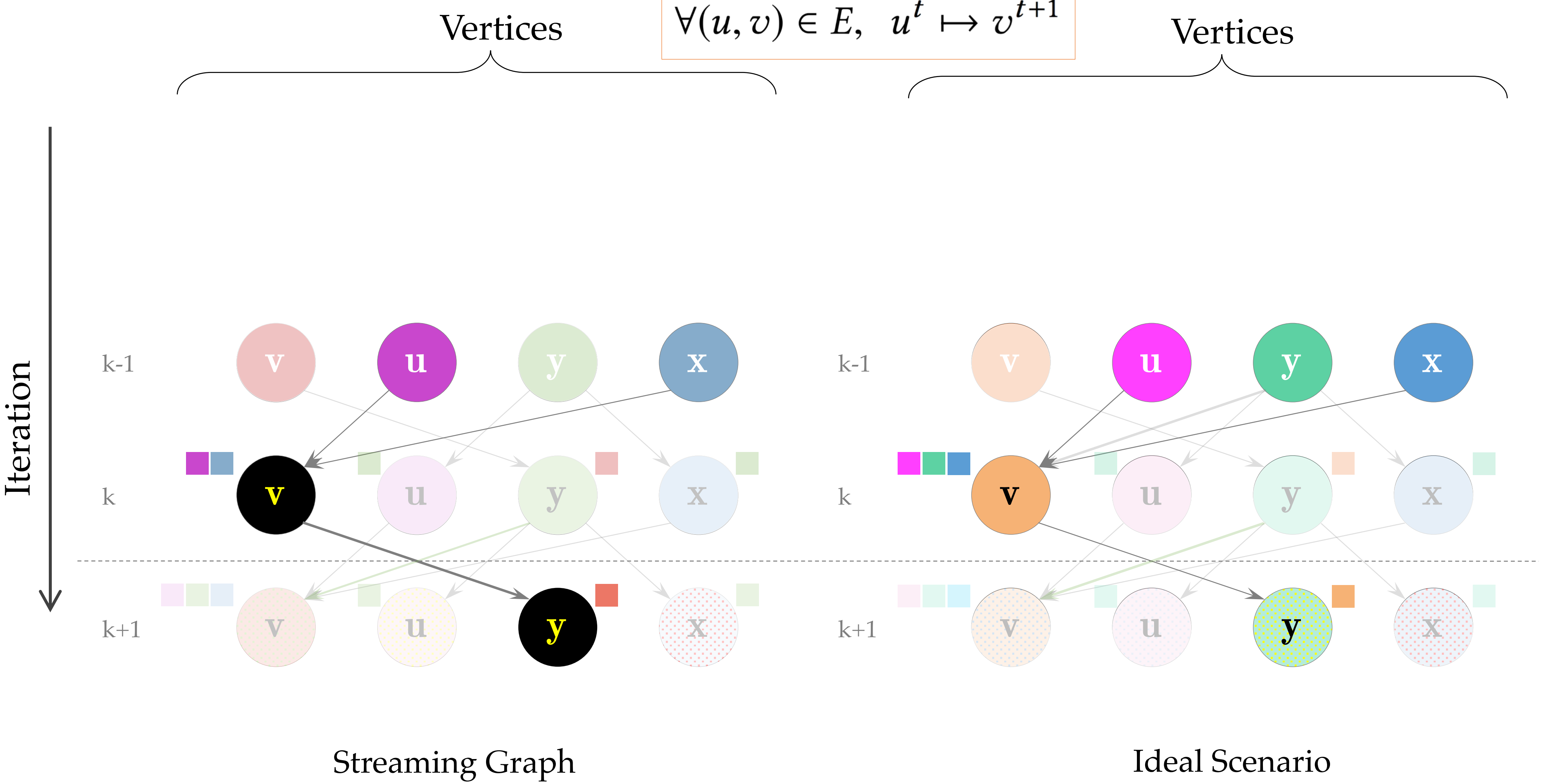
Streaming Graph

Ideal Scenario

Upon Edge Addition

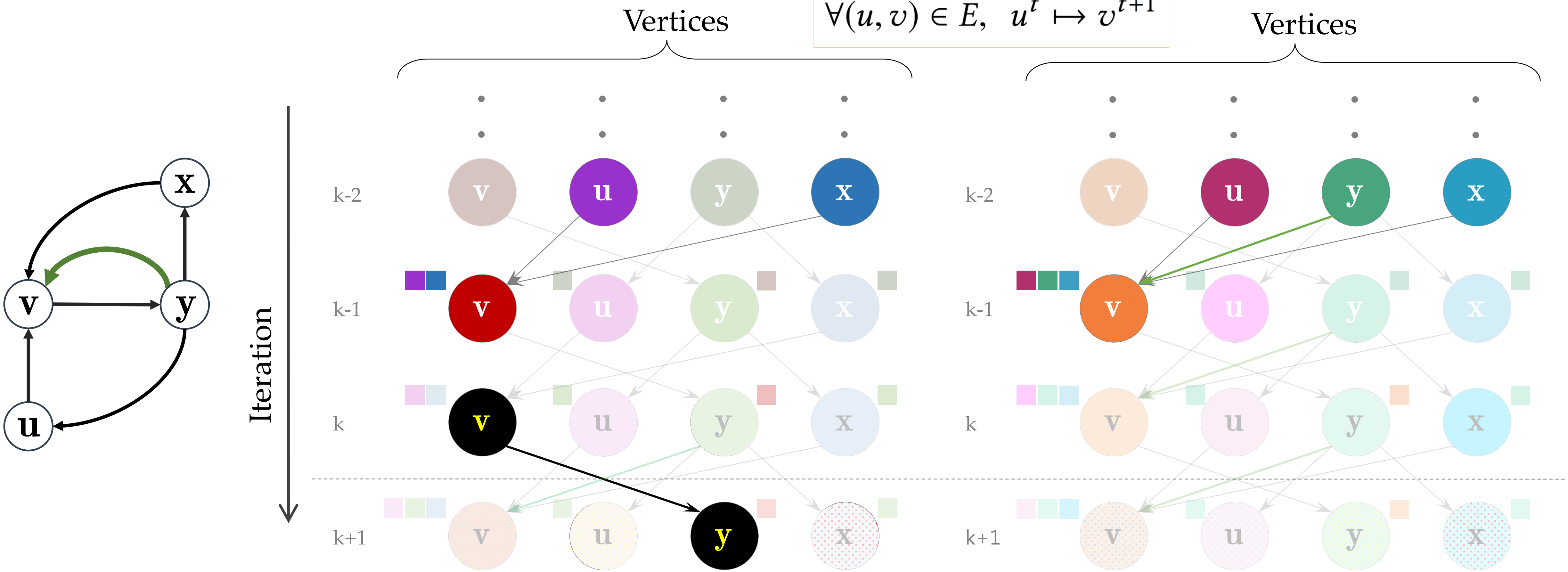
$$\boxed{\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}}$$





Upon Edge Addition

$$\forall (u,v) \in E, \quad u^t \mapsto v^{t+1}$$

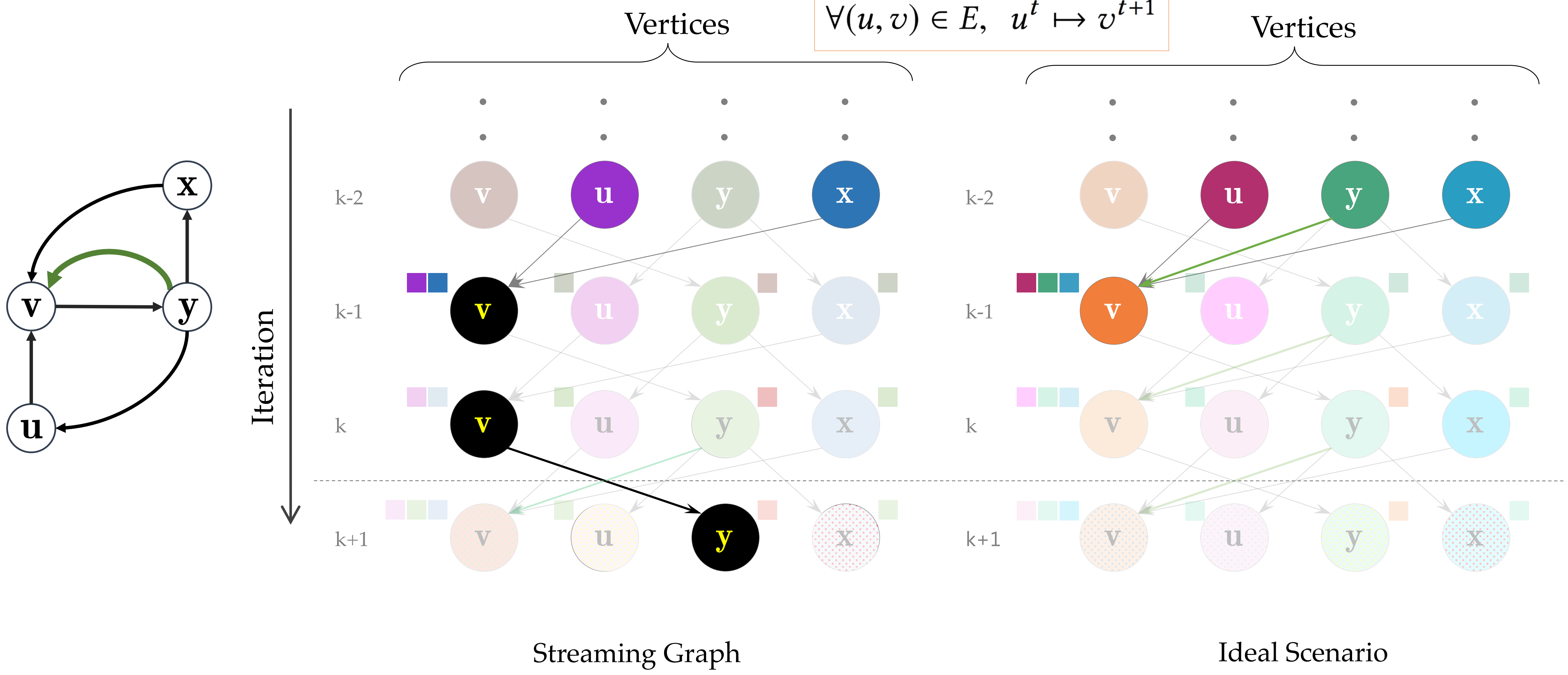


Streaming Graph

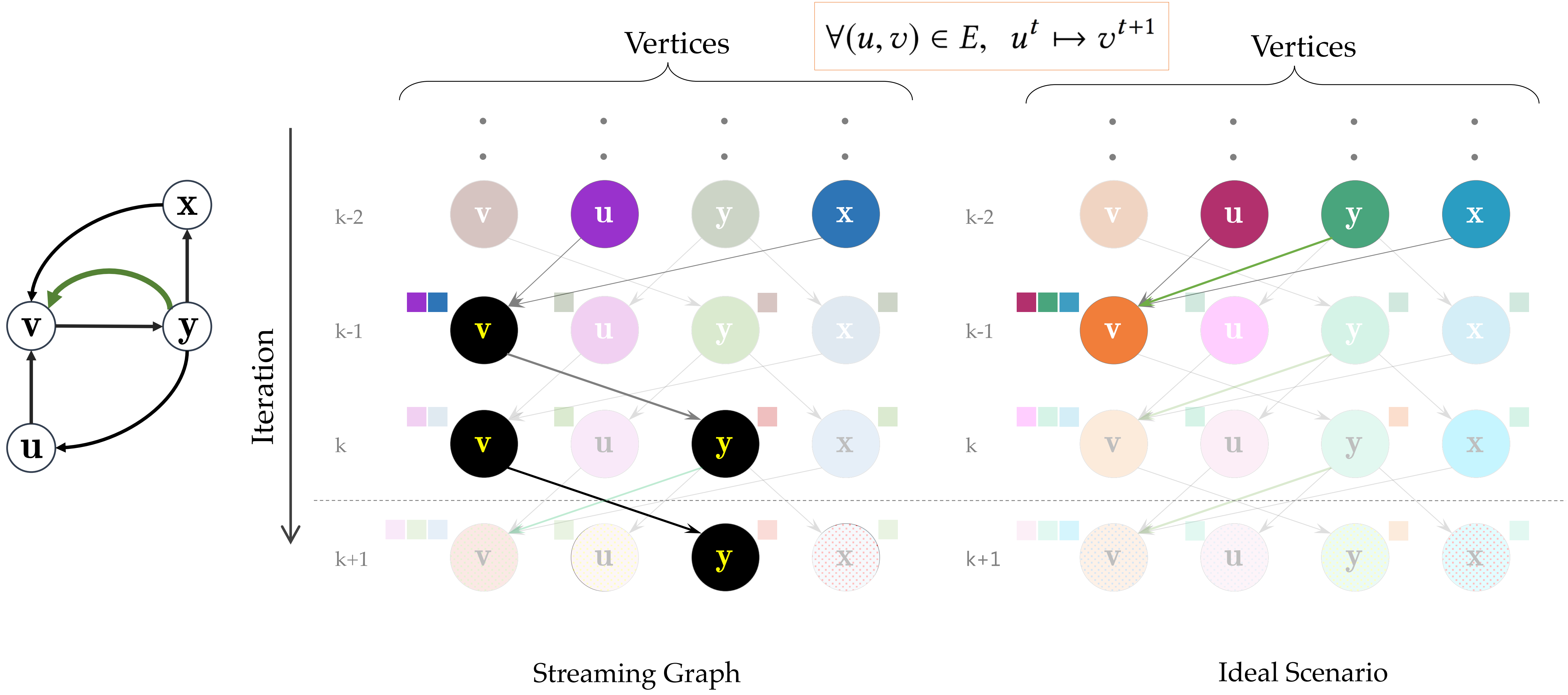
Ideal Scenario

Upon Edge Addition

$$\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}$$

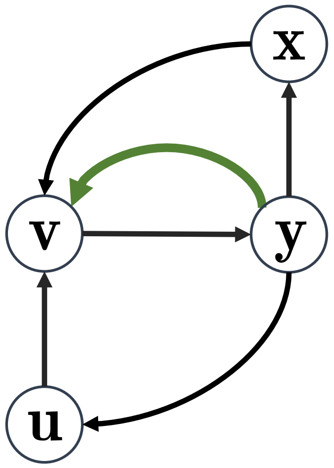


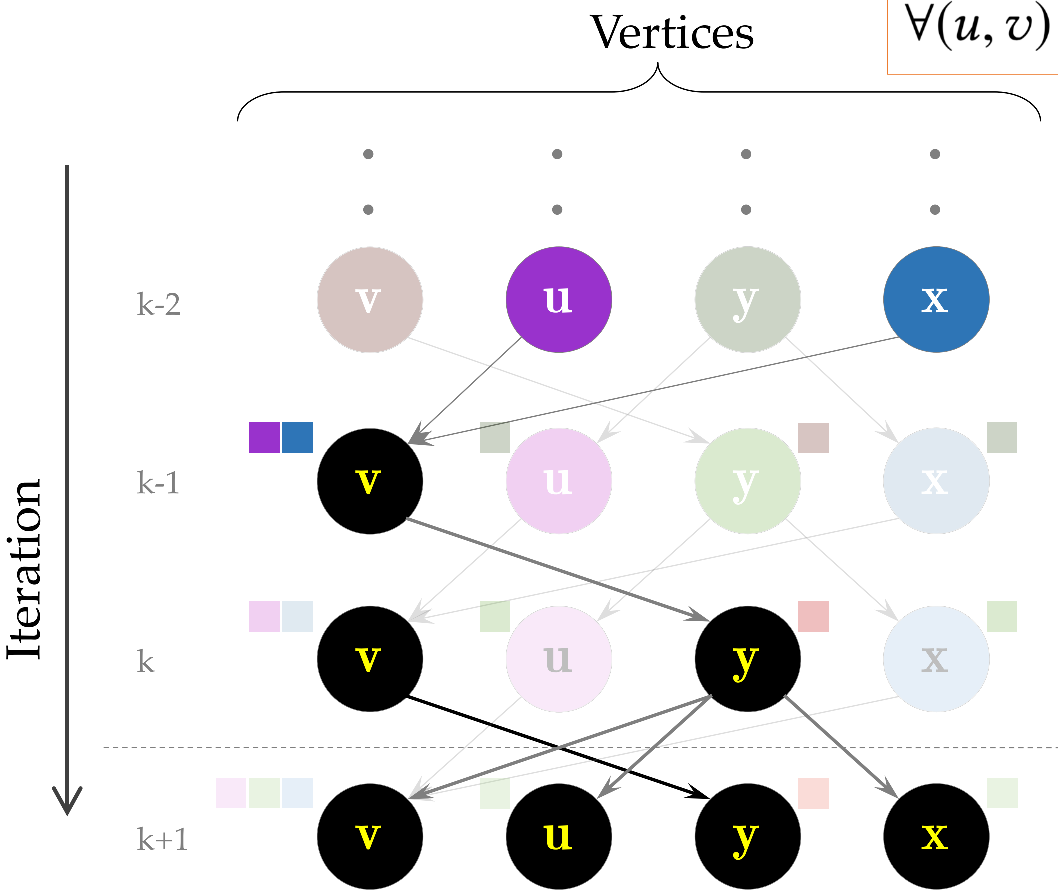
Upon Edge Addition



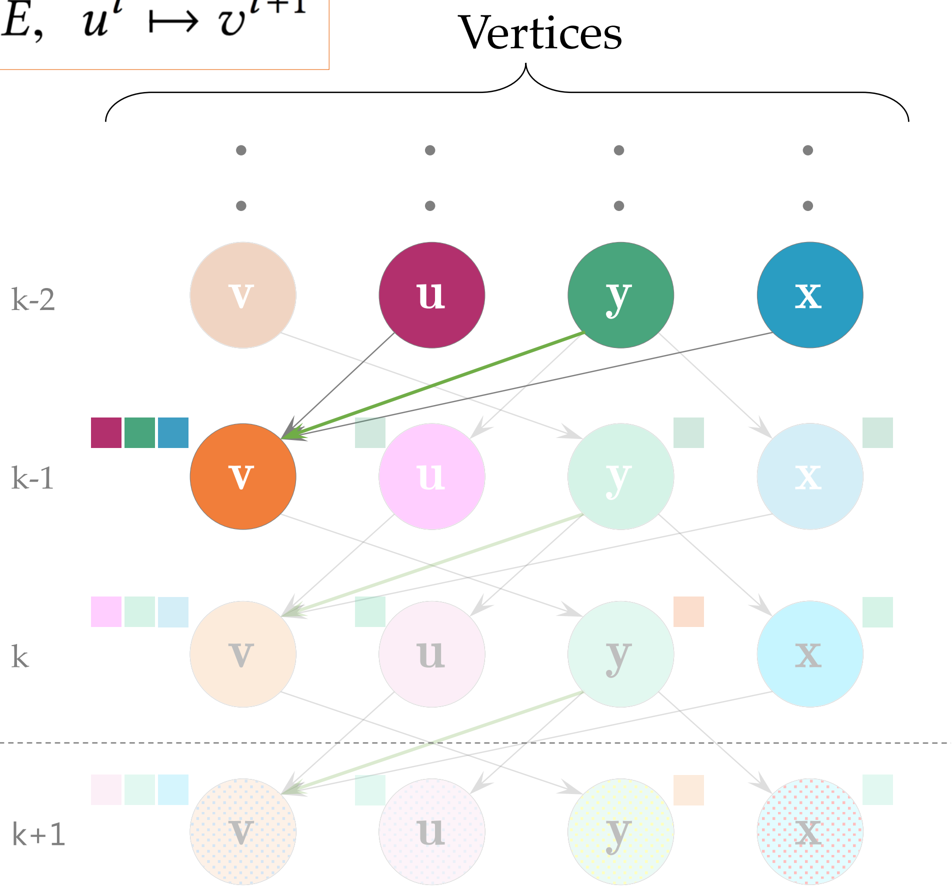
Upon Edge Addition

$$\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}$$





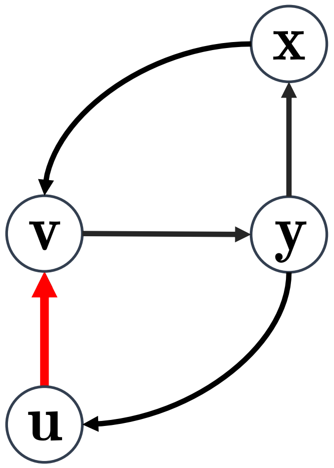
Streaming Graph

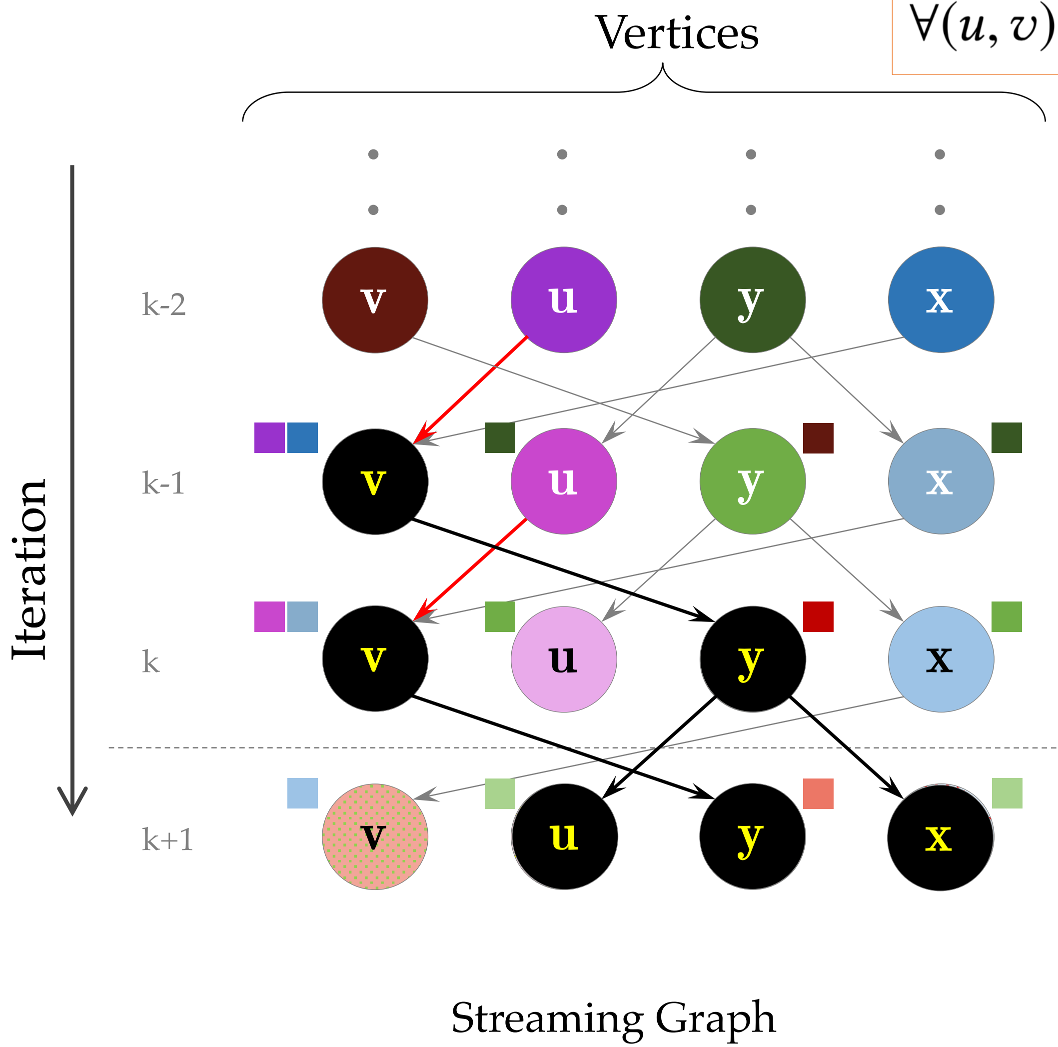


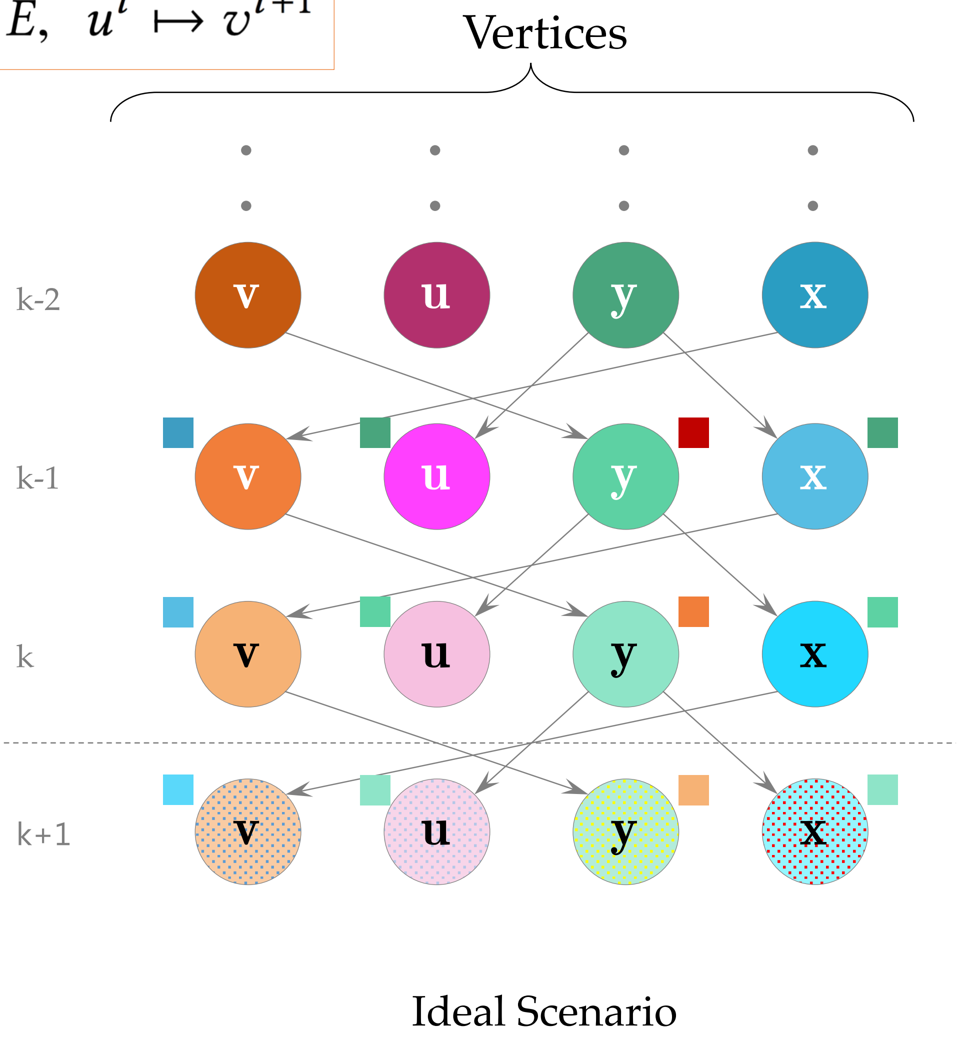
Ideal Scenario

Upon Edge Deletion

$$\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}$$

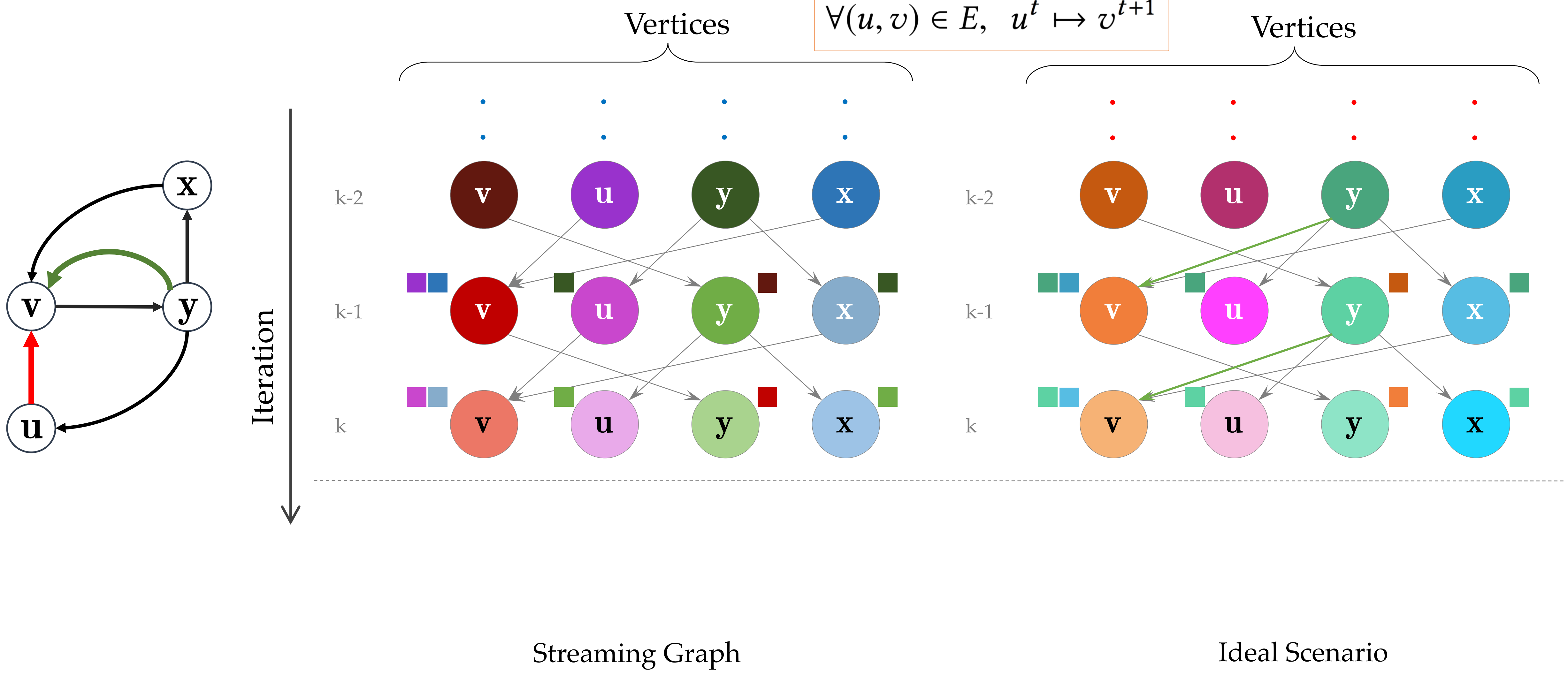






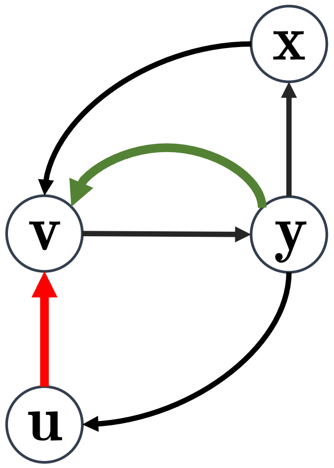
GraphBolt: Correcting Dependencies

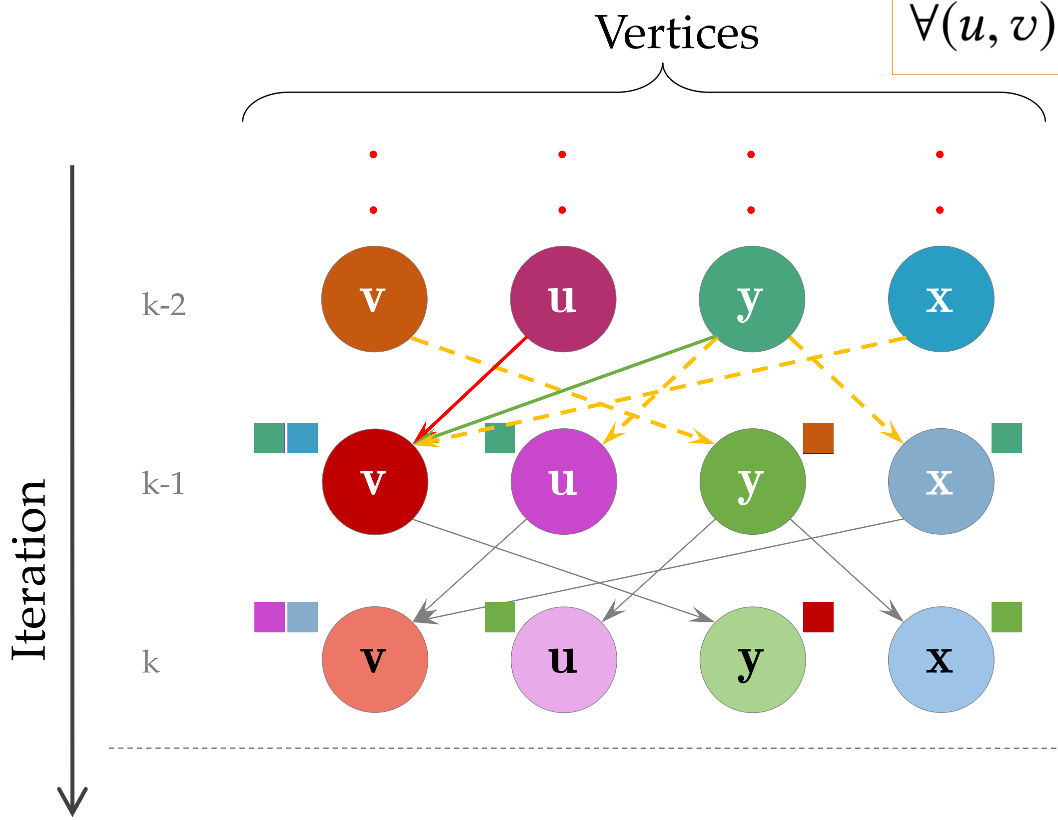
$$\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}$$

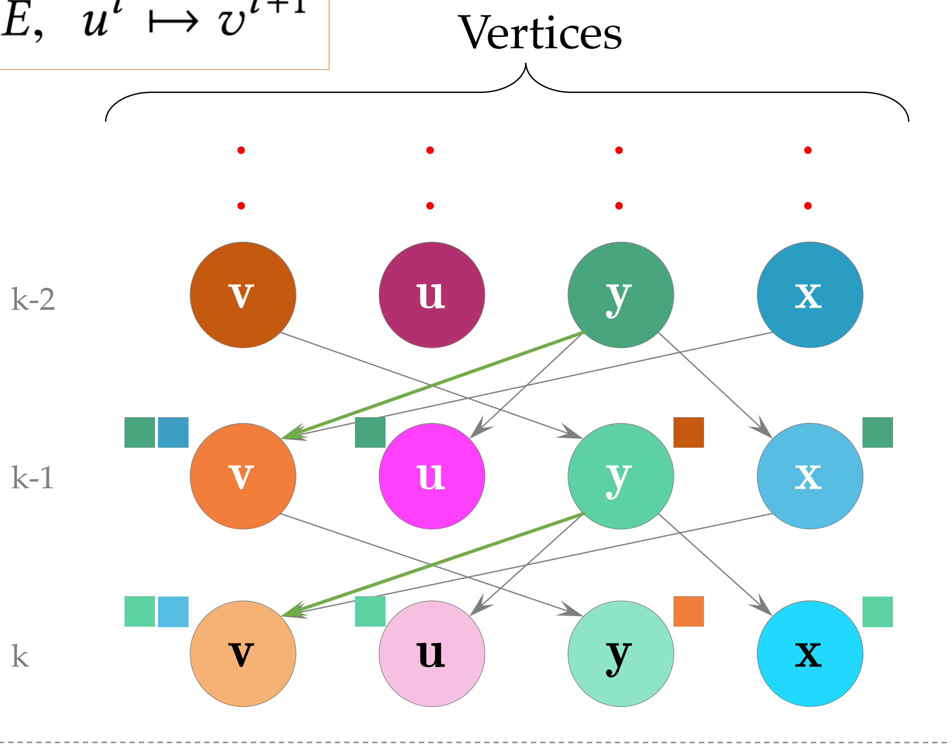


GraphBolt: Correcting Dependencies

$\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}$





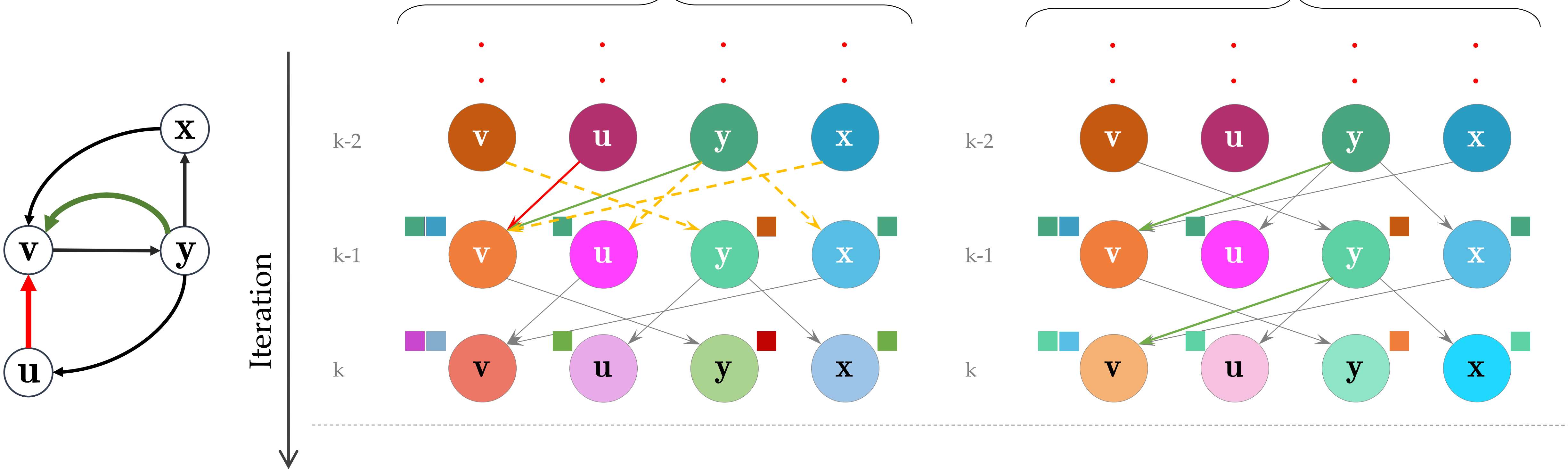


Streaming Graph

Ideal Scenario

GraphBolt: Correcting Dependencies

$$\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}$$

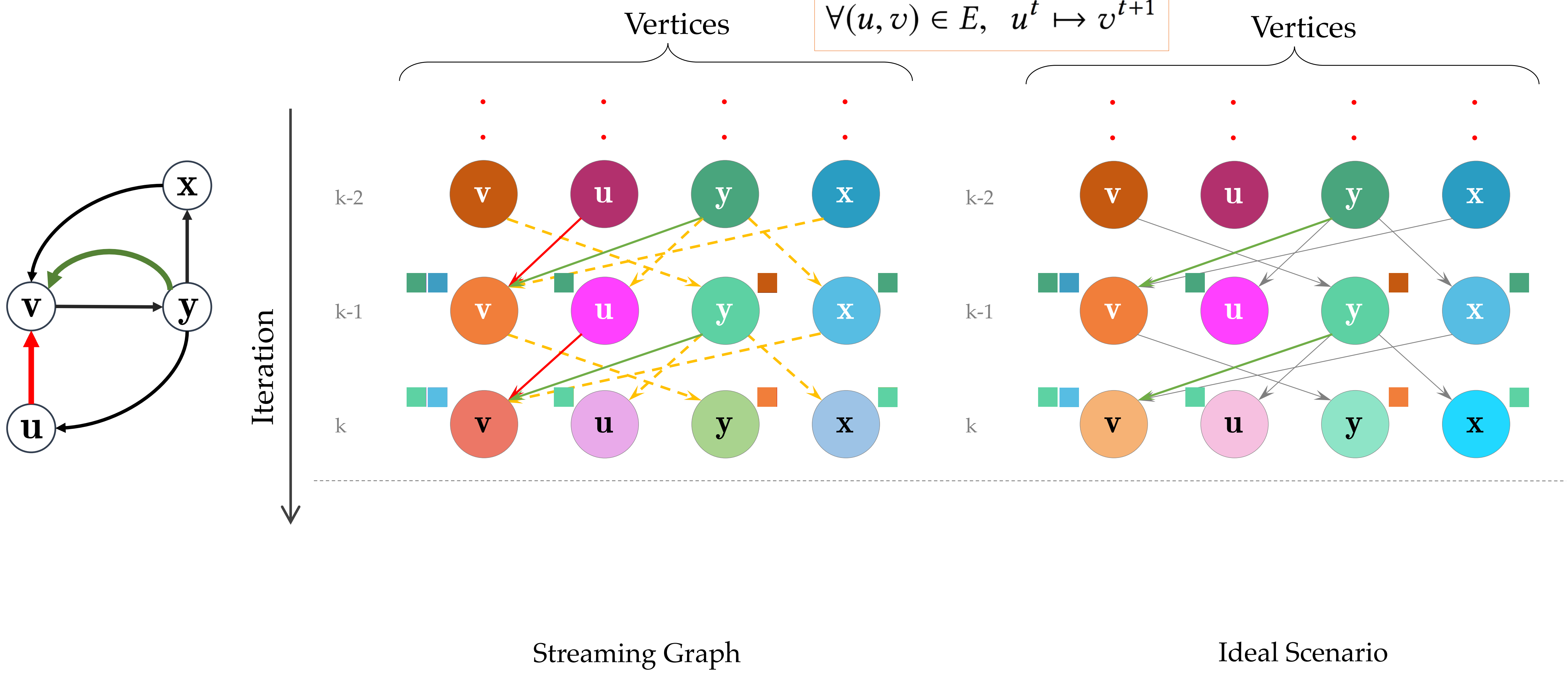


Streaming Graph

Ideal Scenario

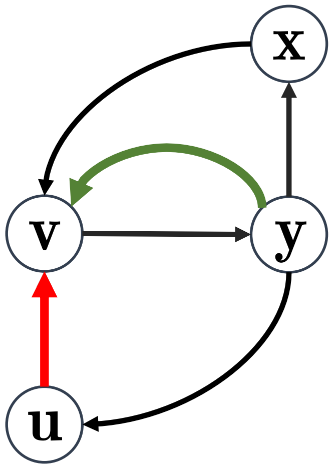
GraphBolt: Correcting Dependencies

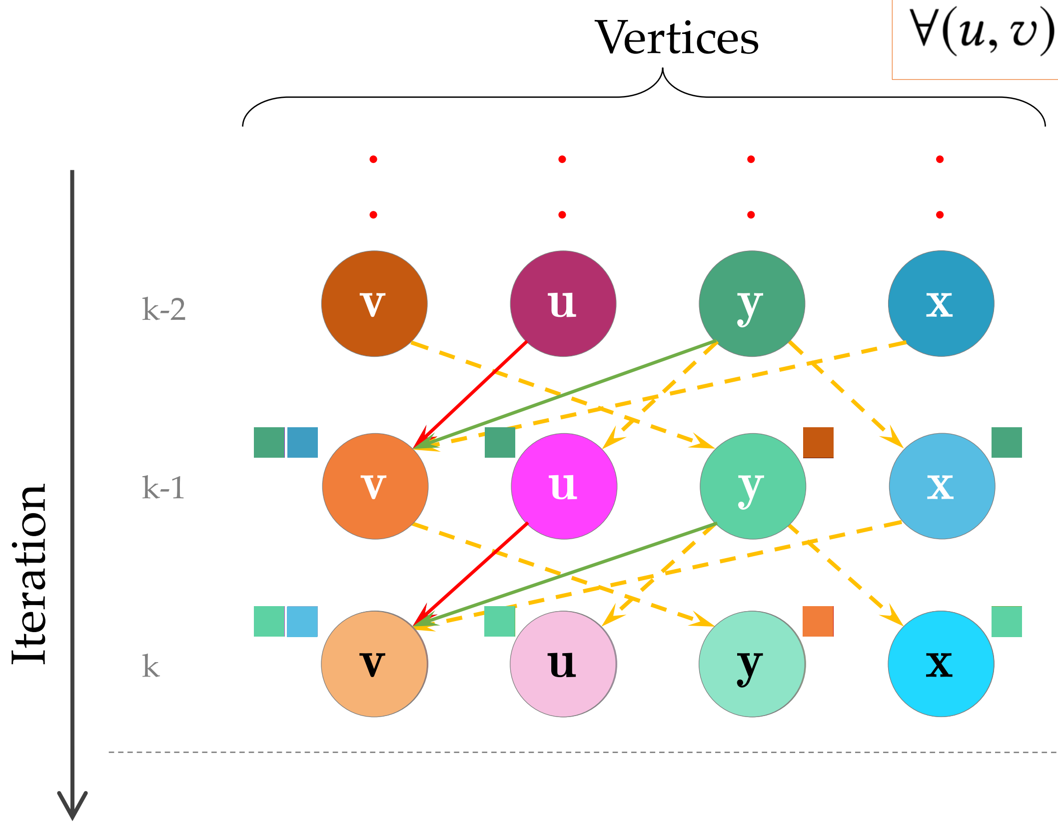
$$\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}$$



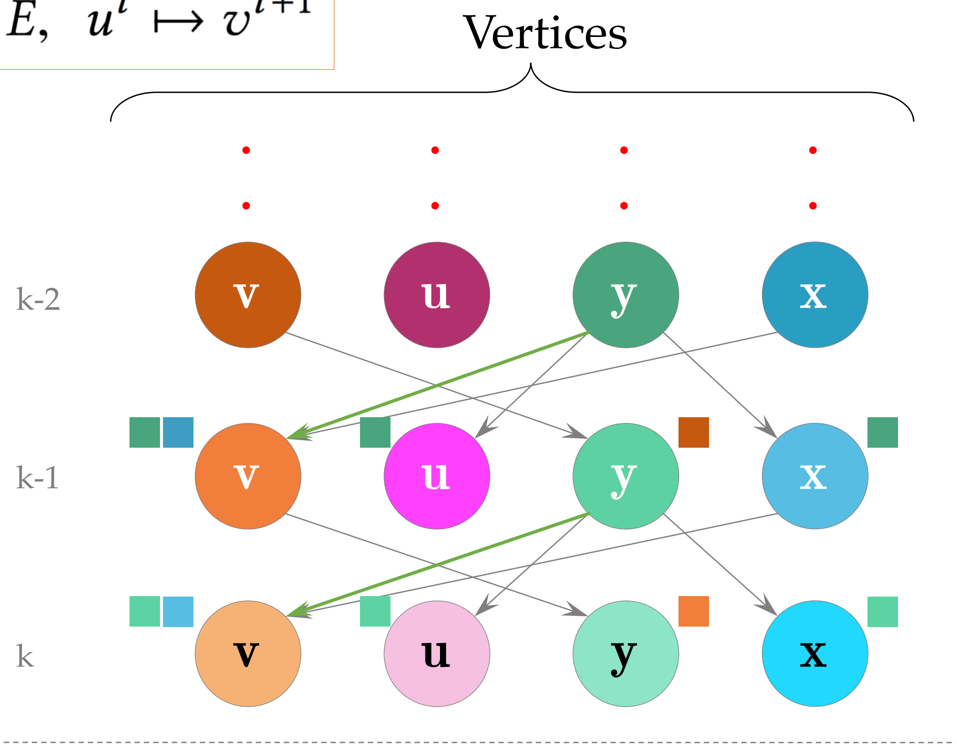
GraphBolt: Correcting Dependencies

$\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}$





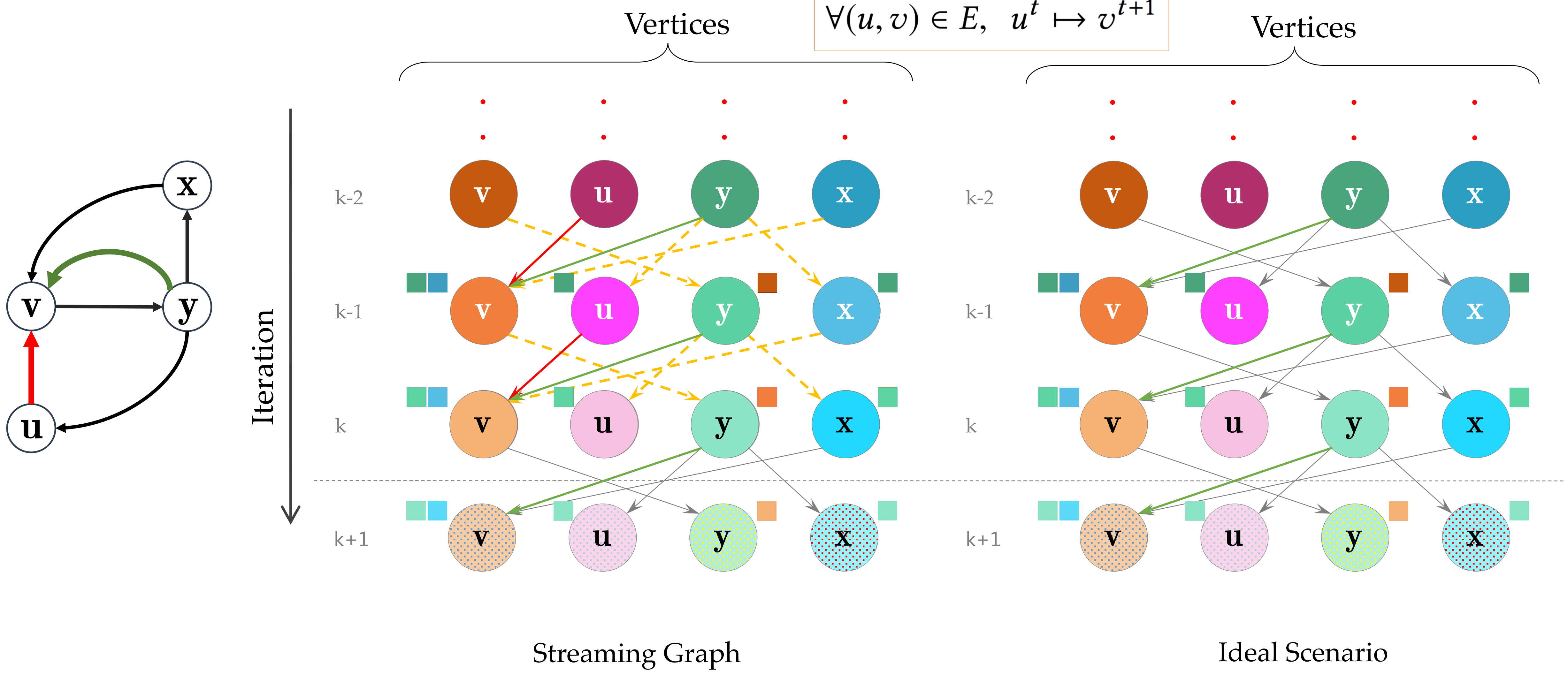
Streaming Graph



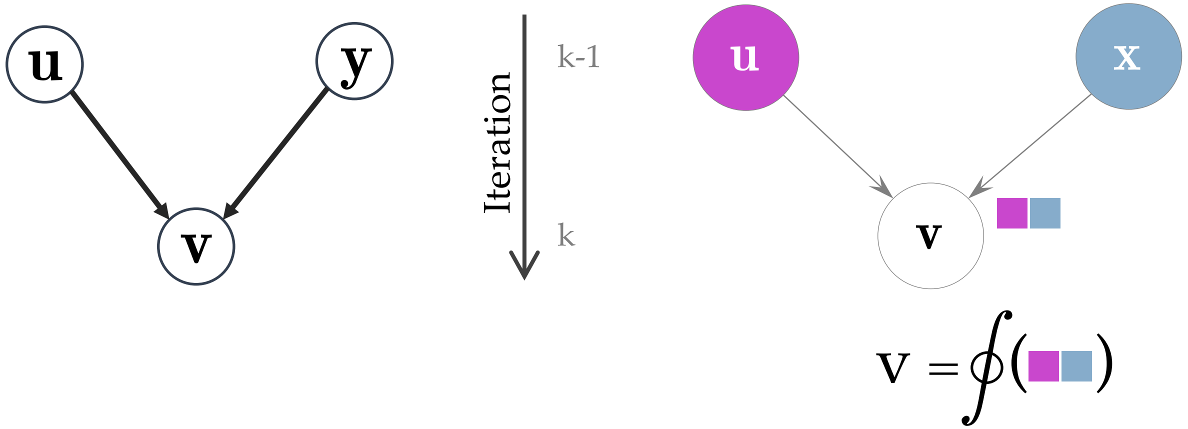
Ideal Scenario

GraphBolt: Correcting Dependencies

$$\forall (u, v) \in E, \quad u^t \mapsto v^{t+1}$$



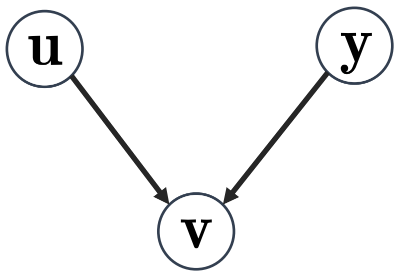
GraphBolt: Dependency Tracking

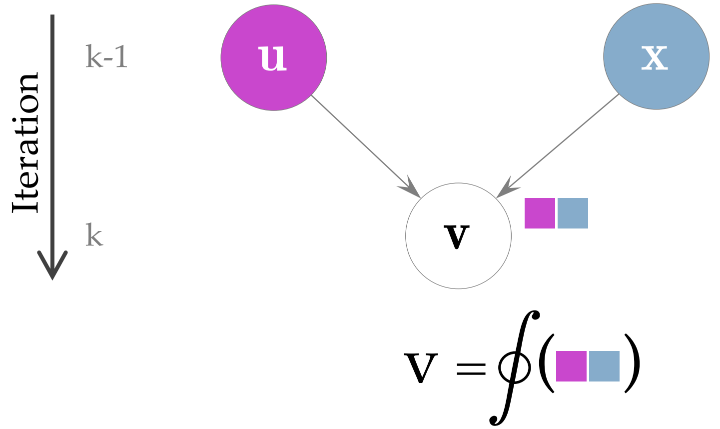


$$c_i(v) = \oint \Big(\bigoplus_{\forall e=(u,v)\in E} (c_{i-1}(u)) \Big)$$

$$g_i(v) = \bigoplus_{\forall e=(u,v)\in E} (c_{i-1}(u))$$

GraphBolt: Dependency Tracking



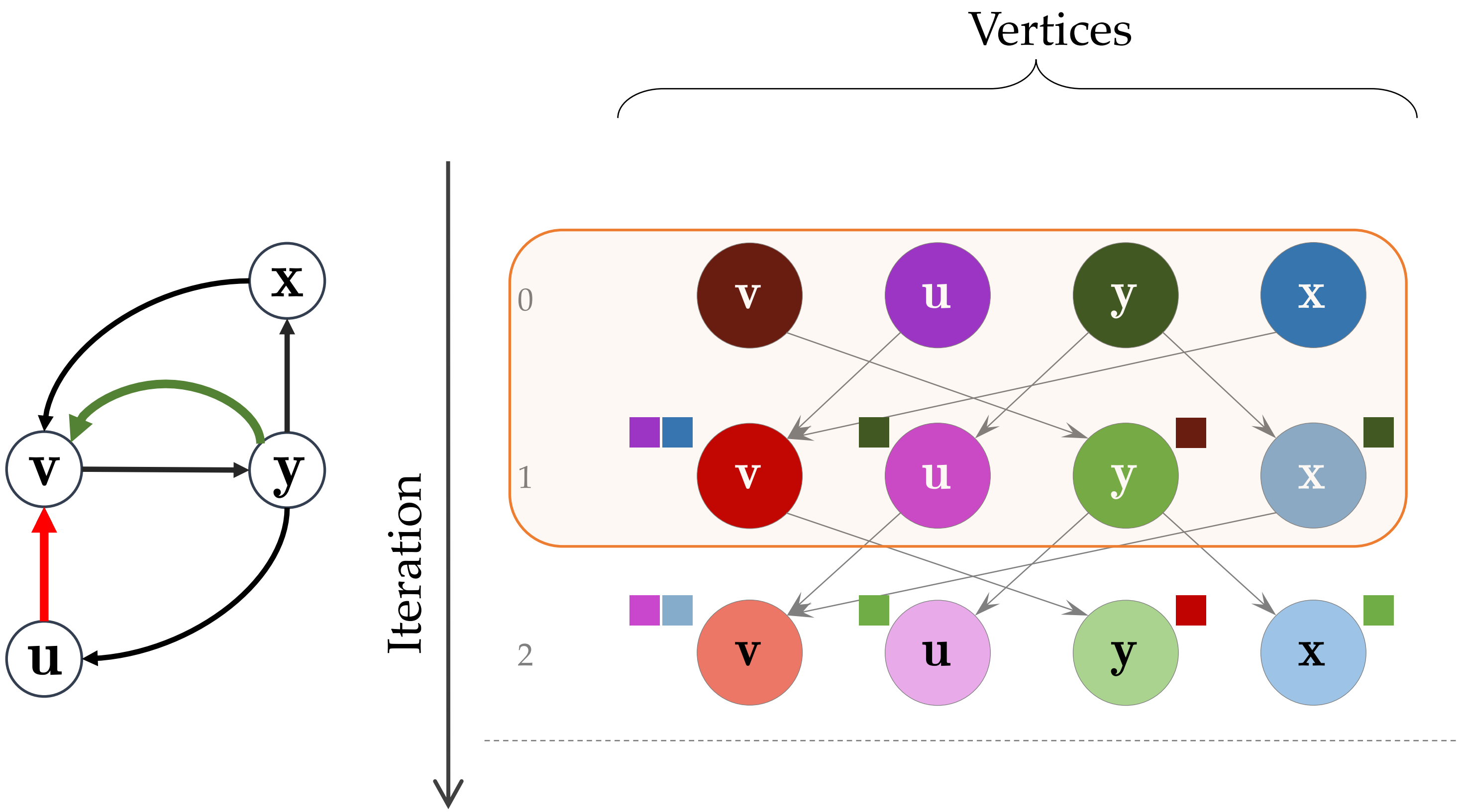


- Dependency relations translate across aggregation points
- Structure of dependencies inferred from input graphs

$$c_i(v) = \oint_{\forall e=(u,v)\in E} \left(\boxed{\bigoplus (c_{i-1}(u))} \right)$$

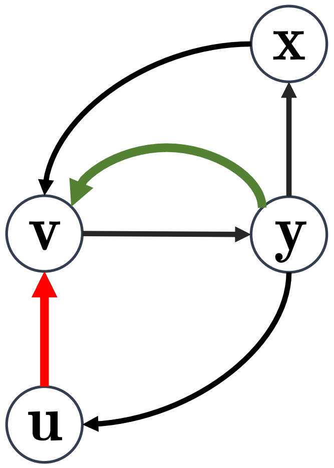
$$g_i(v) = \bigoplus_{\forall e=(u,v)\in E} (c_{i-1}(u))$$

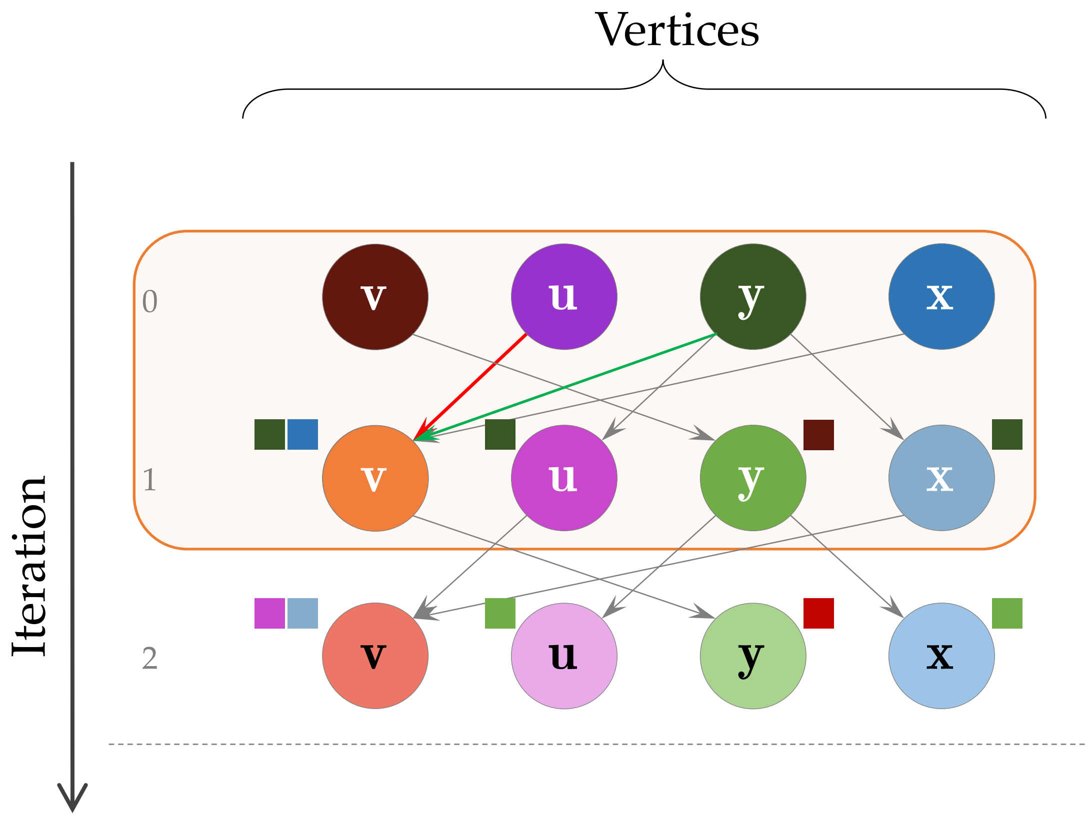
GraphBolt: Incremental Refinement



$$g_i^T(v) = g_i(v) \; + \, ?$$

GraphBolt: Incremental Refinement





Refinement

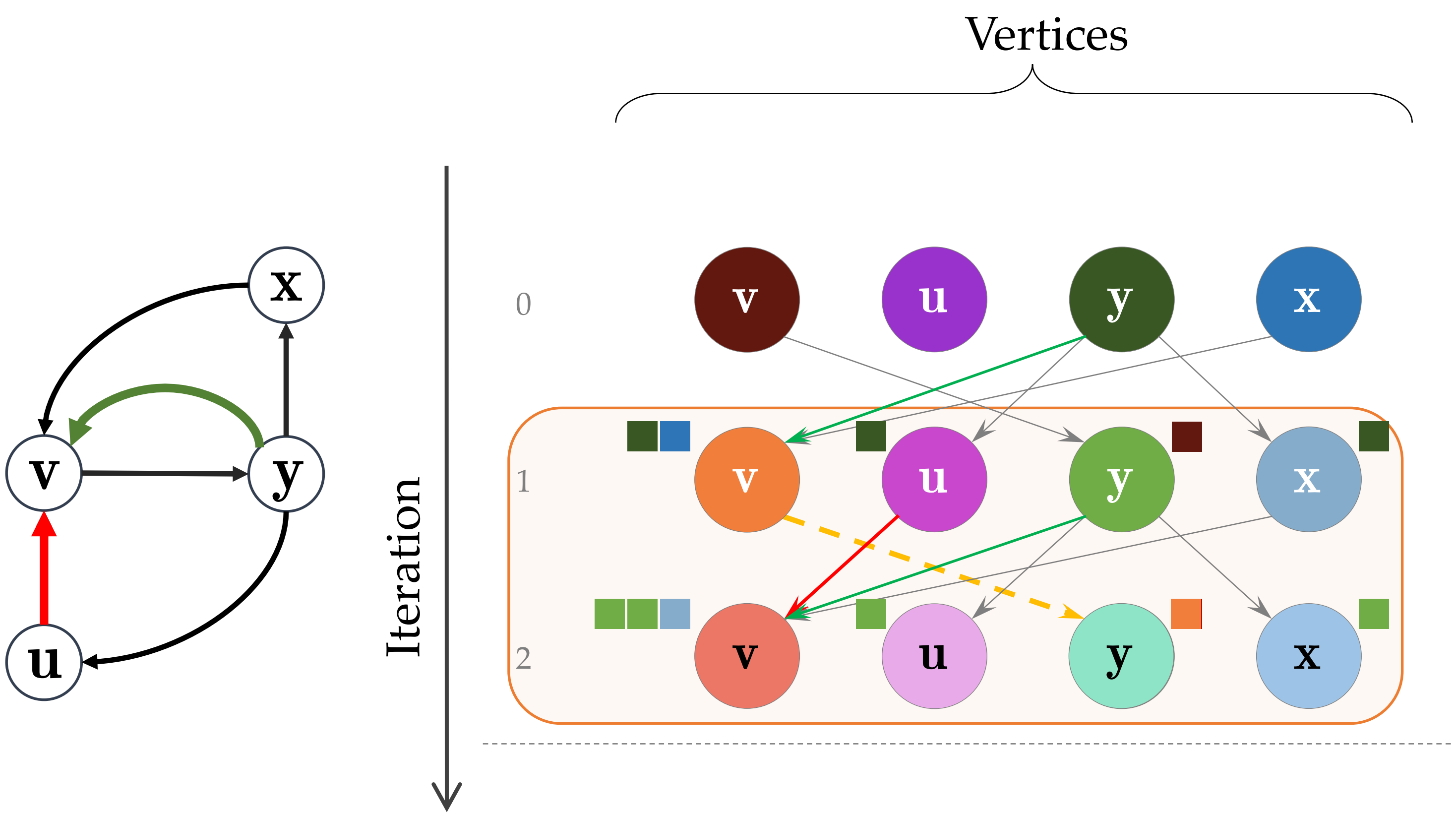
Direct changes | Transitive changes

Edge additions $\underset{\forall e=(u,v)\in E_a}{\biguplus} (c_{i-1}(u))$

Edge deletions $\underset{\forall e=(u,v)\in E_d}{\mathop{\bigcup\!\!\!\!\!\!-\;}} (c_{i-1}(u))$

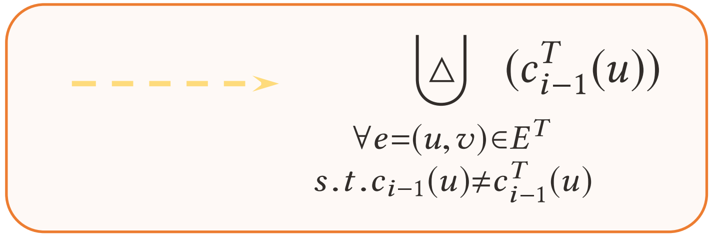
$$g_i^T(v) = g_i(v) \underset{\forall e=(u,v)\in E_a}{\biguplus} (c_{i-1}(u)) \underset{\forall e=(u,v)\in E_d}{\mathop{\bigcup\!\!\!\!\!\!-\;}} (c_{i-1}(u)) \quad + \, ?$$

GraphBolt: Incremental Refinement



Refinement

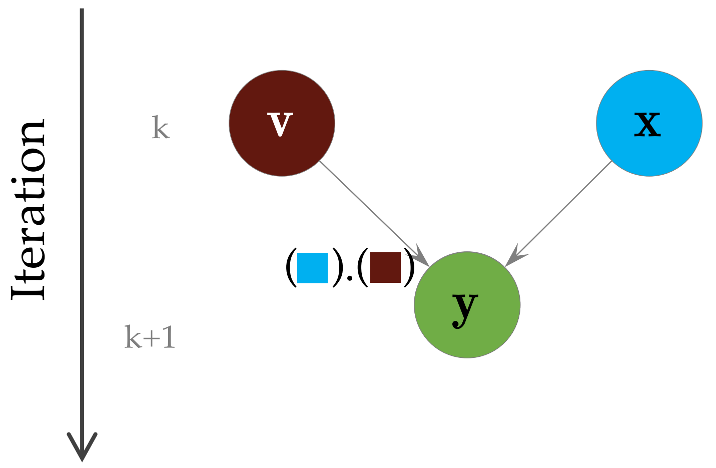
Direct changes | Transitive changes



$$g_i^T(v) = g_i(v) \biguplus_{\forall e=(u,v)\in E_a} (c_{i-1}(u)) \quad \biguplus_{\forall e=(u,v)\in E_d}^{-} (c_{i-1}(u)) \quad \biguplus_{\substack{\forall e=(u,v)\in E^T \\ s.t. c_{i-1}(u)\neq c^T_{i-1}(u)}}^{\triangle} (c^T_{i-1}(u))$$

Refinement: Transitive Changes

Vertex aggregation



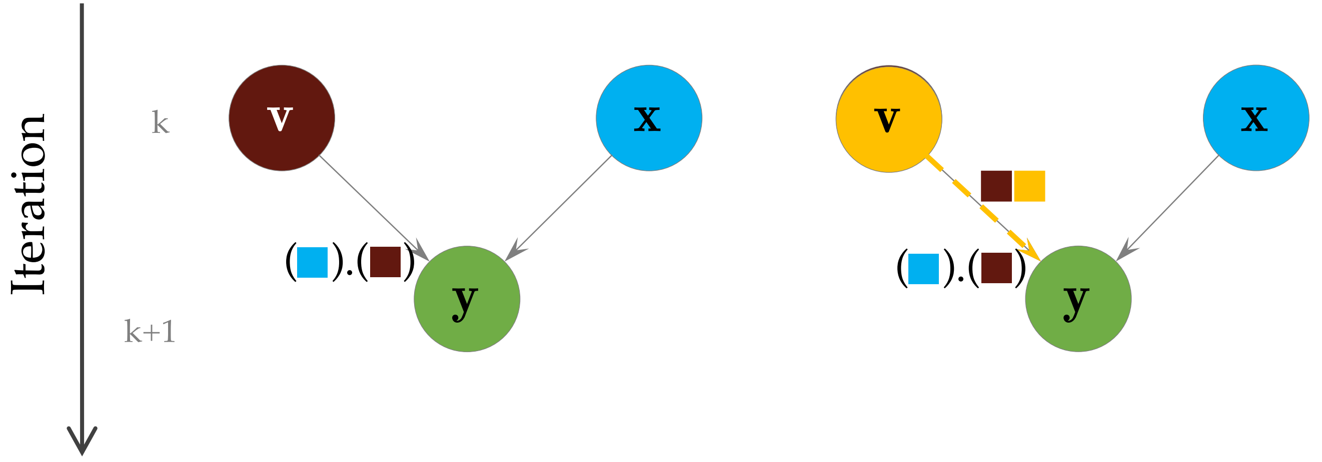
(■) - Transform

(■).(■) - Combine

$$g_i(v) = \bigoplus_{\forall e=(u,v)\in E} (c_{i-1}(u))$$

Refinement: Transitive Changes

Vertex aggregation | Incremental refinement



(■) - Transform

(■).(■) - Combine

$$g_i(v) = \bigoplus_{\forall e=(u,v)\in E} (c_{i-1}(u))$$

Aggregation

Complex

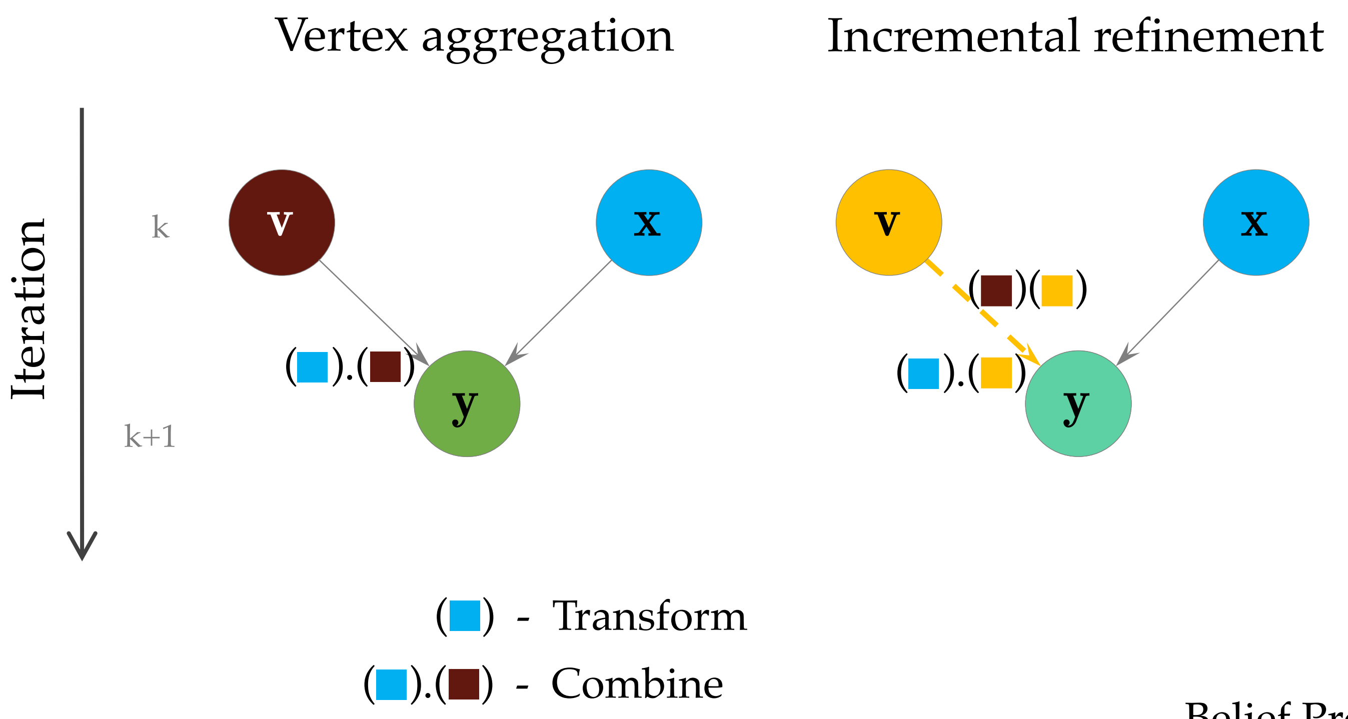
- Retract : Old value
- Propagate : New value

Belief Propagation

$$\forall s \in S : \prod_{\forall e=(u,v)\in E} \left(\sum_{\forall s' \in S} \phi(u,s') \times \psi(u,v,s',s) \times \frac{c(u,s')}{c(v,s')} \right)$$

Aggregation | Transform | Vertex Value

Refinement: Transitive Changes



$$g_i(v) = \bigoplus_{\forall e=(u,v)\in E} (c_{i-1}(u))$$

Aggregation

Complex

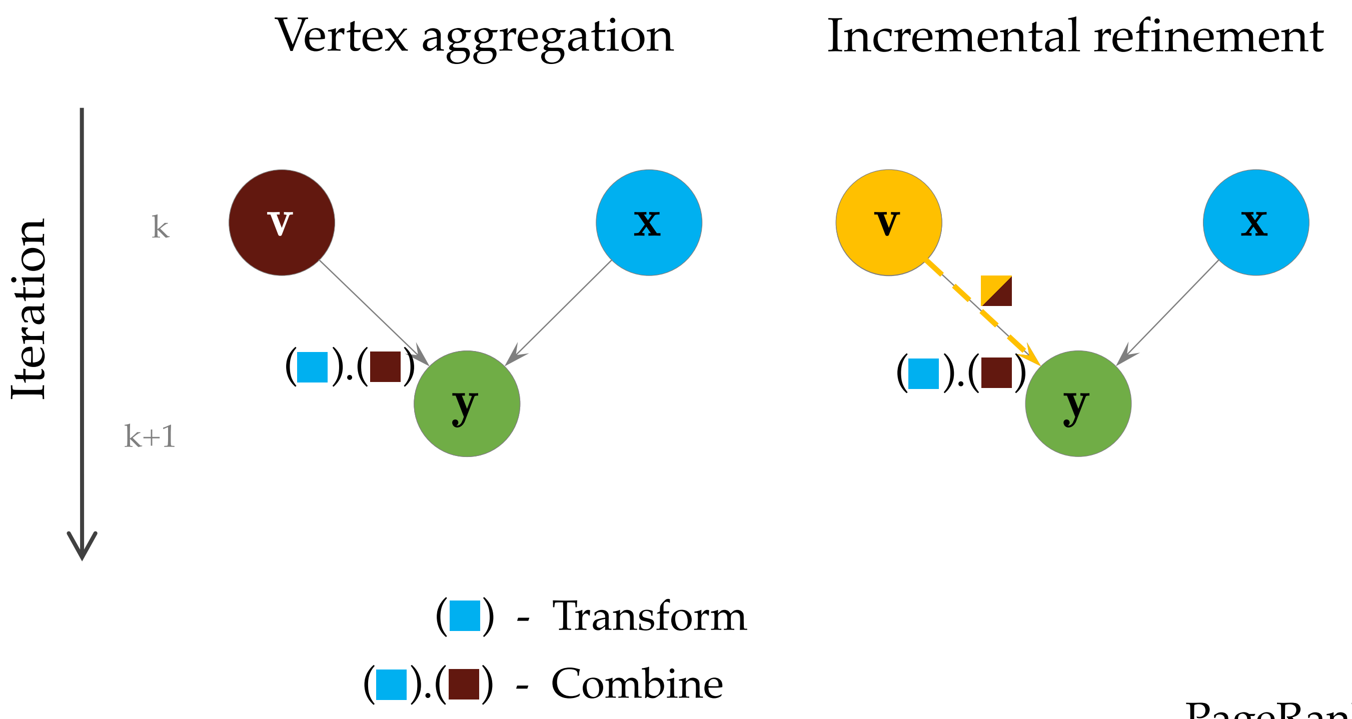
- Retract : Old value
- Propagate : New value

Belief Propagation

$$\forall s \in S : \prod_{\forall e=(u,v)\in E} \left(\sum_{\forall s'\in S} \phi(u,s') \times \psi(u,v,s',s) \times \boxed{\frac{c(u,s')}{c(v,s')}} \right)$$

Aggregation   Transform   Vertex Value

Refinement: Aggregation Types



$$g_i(v) = \bigoplus_{\forall e=(u,v)\in E} (c_{i-1}(u))$$

Aggregation

Complex | Simple

Propagate : Change in vertex value

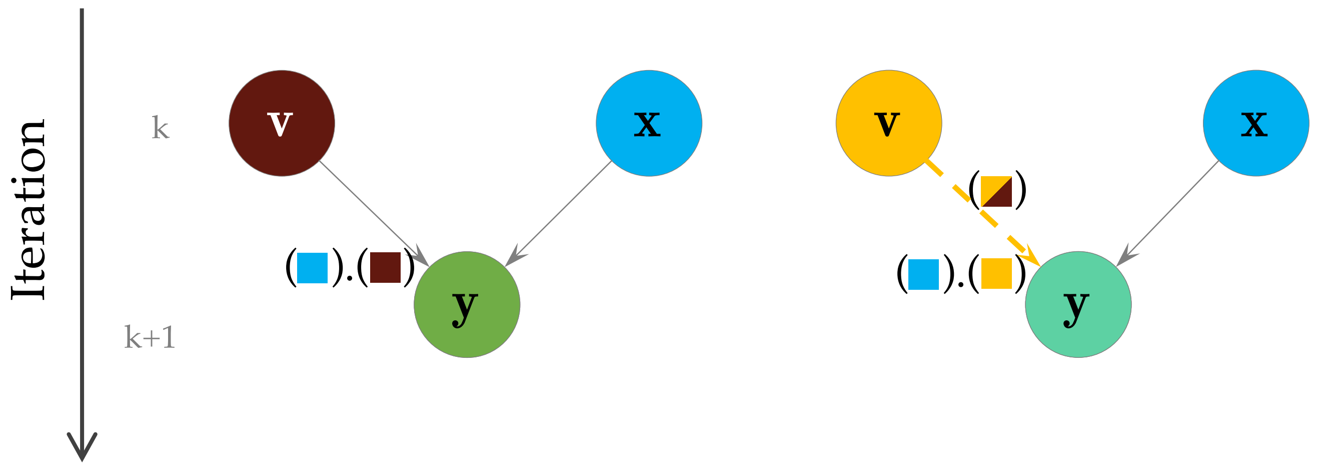
PageRank

$$\sum_{\forall e=(u,v)\in E} \frac{c(u)}{out\_degree(u)}$$

Aggregation Transform Vertex Value

Refinement: Aggregation Types

Vertex aggregation | Incremental refinement



() - Transform

().() - Combine

$$g_i(v) = \bigoplus_{\forall e=(u,v)\in E} (c_{i-1}(u))$$

Aggregation

Complex | **Simple**

Propagate : Change in vertex value

PageRank

$$\sum_{\forall e=(u,v)\in E} \boxed{\frac{\boxed{c(u)}}{out\_degree(u)}}$$

Aggregation Transform Vertex Value

GraphBolt: Programming Model

function REPROPAGATE($e = (u, v),\ i$)
  ATOMICADD($\&sum[v][i+1], \frac{oldpr[u][i]}{old\_degree[u]}$)
end function

function RETRACT($e = (u, v),\ i$)
  ATOMICSUB($\&sum[v][i+1], \frac{oldpr[u][i]}{old\_degree[u]}$)
end function

function PROPAGATE($e = (u, v),\ i$)
  ATOMICADD($\&sum[v][i+1], \frac{newpr[u][i]}{new\_degree[u]}$)
end function

function PAGERANK( )
  for $i \in [0 \ldots k]$ **do**
    EDGEMAP(E_add, REPROPAGATE, i)
    EDGEMAP(E_delete, RETRACT, i)
  end for
  $V\_updated$ = GETSOURCES($E\_add \cup E\_delete$)
  V_change = GETTARGETS($E\_add \cup E\_delete$)
  for $i \in [0 \ldots k]$ **do**
    $E\_update = \{(u, v) : u \in V\_updated\}$
    EDGEMAP(E_update, RETRACT, i)
    EDGEMAP(E_update, PROPAGATE, i)
    V_dest = GETTARGETS(E_update)
    $V\_change = V\_change \cup V\_dest$
    $V\_updated$ = VERTEXMAP(V_change, COMPUTE, i)
  end for
end function

Direct changes

GraphBolt: Programming Model - Complex aggregations

```
function REPROPAGATE(e = (u, v), i)
    ATOMICADD(&sum[v][i + 1], oldpr[u][i] / old_degree[u])
end function
```

```
function RETRACT(e = (u, v), i)
    ATOMICSUB(&sum[v][i + 1], oldpr[u][i] / old_degree[u])
end function
```

```
function PROPAGATE(e = (u, v), i)
    ATOMICADD(&sum[v][i + 1], newpr[u][i] / new_degree[u])
end function
```

```
function PAGERANK( )
    for i ∈ [0...k] do
        EDGEMAP(E_add, REPROPAGATE, i)
        EDGEMAP(E_delete, RETRACT, i)
    end for
    V_updated = GETSOURCES(E_add ∪ E_delete)
    V_change = GETTARGETS(E_add ∪ E_delete)
    for i ∈ [0...k] do
        E_update = {(u, v) : u ∈ V_updated}
        EDGEMAP(E_update, RETRACT, i)
        EDGEMAP(E_update, PROPAGATE, i)
        V_dest = GETTARGETS(E_update)
        V_change = V_change ∪ V_dest
        V_updated = VERTEXMAP(V_change, COMPUTE, i)
    end for
end function
```

Transitive changes

GraphBolt: Programming Model - Simple aggregations

function REPROPAGATE$(e = (u, v),\ i)$
$$\text{ATOMICADD}(\&sum[v][i+1], \frac{oldpr[u][i]}{old\_degree[u]})$$
end function

function RETRACT$(e = (u, v),\ i)$
$$\text{ATOMICSUB}(\&sum[v][i+1], \frac{oldpr[u][i]}{old\_degree[u]})$$
end function

function PROPAGATEDELTA$(e = (u, v),\ i)$
$$\text{ATOMICADD}(\&sum[v][i+1], \frac{newpr[u][i]}{new\_degree[u]} - \frac{oldpr[u][i]}{old\_degree[u]})$$
end function

function PAGERANK( )
  for $i \in [0...k]$ **do**
    EDGEMAP(E_add, REPROPAGATE, i)
    EDGEMAP(E_delete, RETRACT, i)
  end for
  $V\_updated$ = GETSOURCES($E\_add \cup E\_delete$)
  V_change = GETTARGETS($E\_add \cup E\_delete$)
  for $i \in [0...k]$ **do**
    $E\_update = \{(u, v) : u \in V\_updated\}$ Transitive changes
    EDGEMAP(E_update, PROPAGATEDELTA, i)
    V_dest = GETTARGETS(E_update)
    $V\_change = V\_change \cup V\_dest$
    $V\_updated$ = VERTEXMAP(V_change, COMPUTE, i)
  end for
end function

GraphBolt Details …

- **Aggregation properties** & **non-decomposable** aggregations
- **Pruning dependencies** for light-weight tracking
  - Vertical pruning
  - Horizontal pruning
- **Hybrid incremental** execution
  - With & without dependency information
- Graph data structure & parallelization model

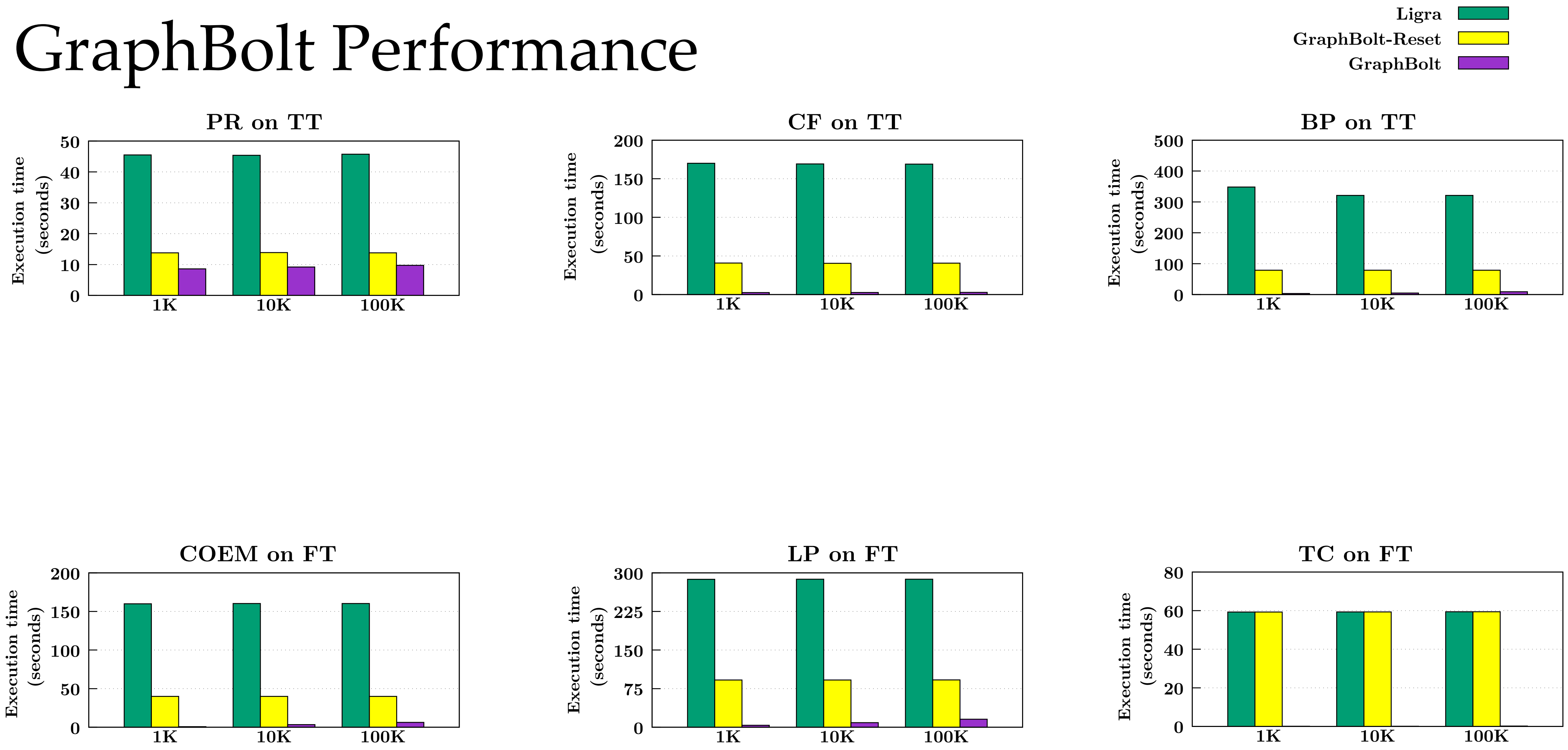
Experimental Setup

Algorithm	Aggregation ($\bigoplus$)
PageRank (PR)	$\sum\limits_{\forall e=(u,v)\in E} \frac{c(u)}{out\_degree(u)}$
Belief Propagation (BP)	$\forall s \in S : \prod\limits_{\forall e=(u,v)\in E} (\sum\limits_{\forall s'\in S} \phi(u,s') \times \psi(u,v,s',s) \times c(u,s'))$
Label Propagation (LP)	$\forall f \in F : \sum\limits_{\forall e=(u,v)\in E} c(u,f) \times weight(u,v)$
Co-Training Expectation Maximization (CoEM)	$\sum\limits_{\forall e=(u,v)\in E} \frac{c(u)\times weight(u,v)}{\sum\limits_{\forall e=(w,v)\in E} weight(w,v)}$
Collaborative Filtering (CF)	$\langle \sum_{\forall e=(u,v)\in E} c_i(u).c_i(u)^{tr} , \sum_{\forall e=(u,v)\in E} c_i(u).weight(u,v) \rangle$
Triangle Counting (TC)	$\sum\limits_{\forall e=(u,v)\in E} \lvert in\_neighbors(u) \cap out\_neighbors(v) \rvert$

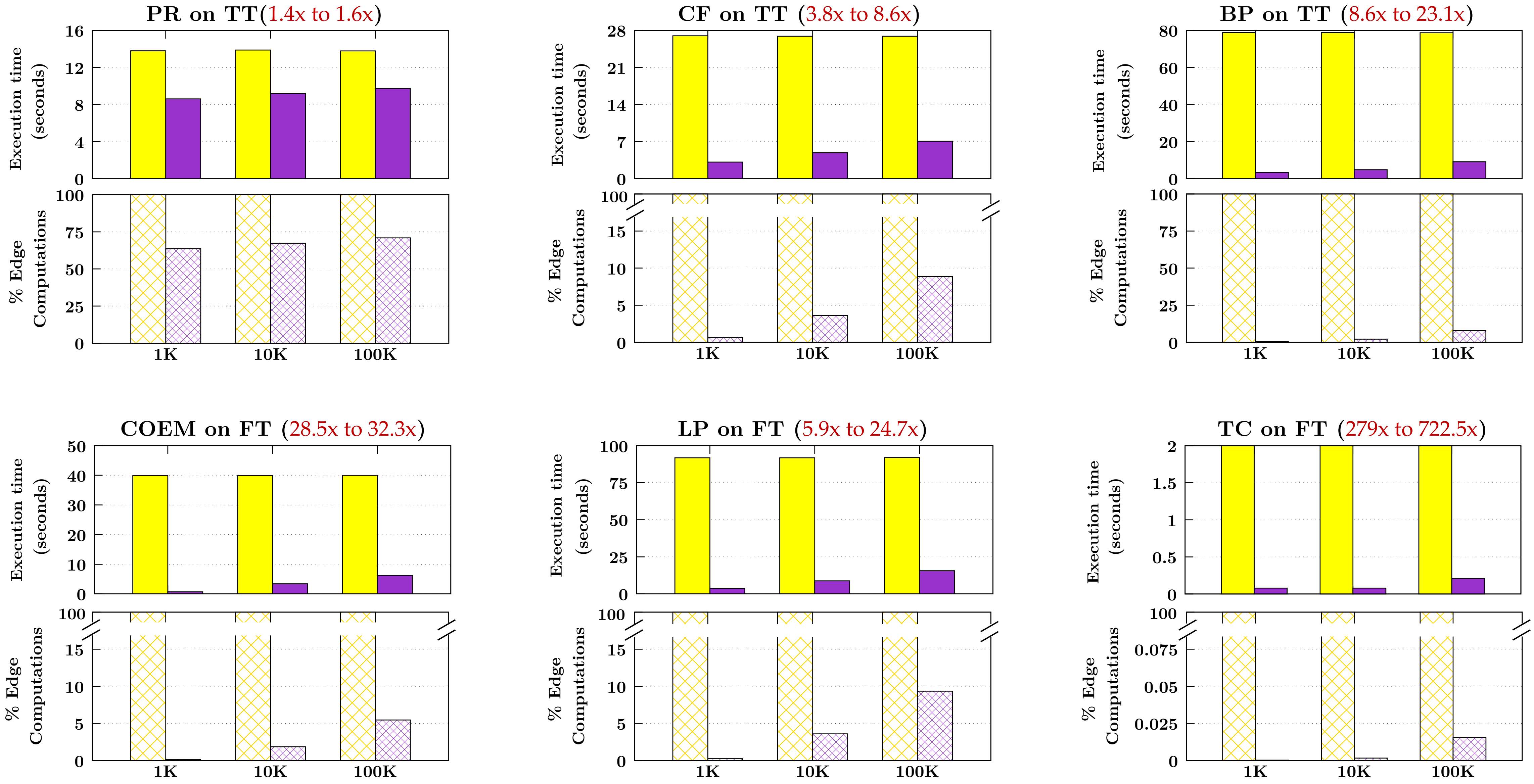
Graphs	Edges	Vertices
Wiki (WK) [47]	378M	12M
UKDomain (UK) [7]	1.0B	39.5M
Twitter (TW) [21]	1.5B	41.7M
TwitterMPI (TT) [8]	2.0B	52.6M
Friendster (FT) [14]	2.5B	68.3M
Yahoo (YH) [49]	6.6B	1.4B

Server: 32-core / 2 GHz / 231 GB

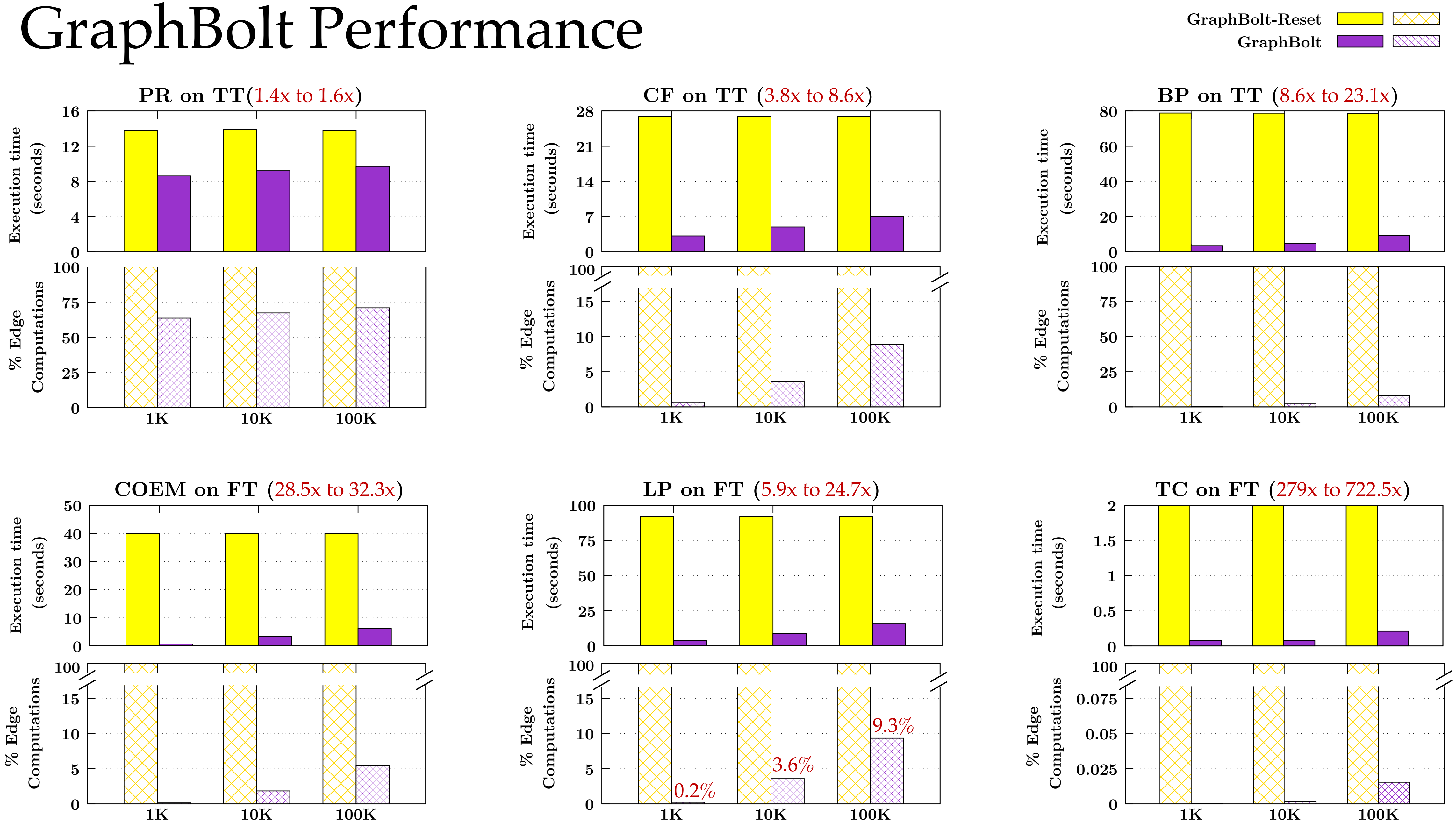
GraphBolt Performance



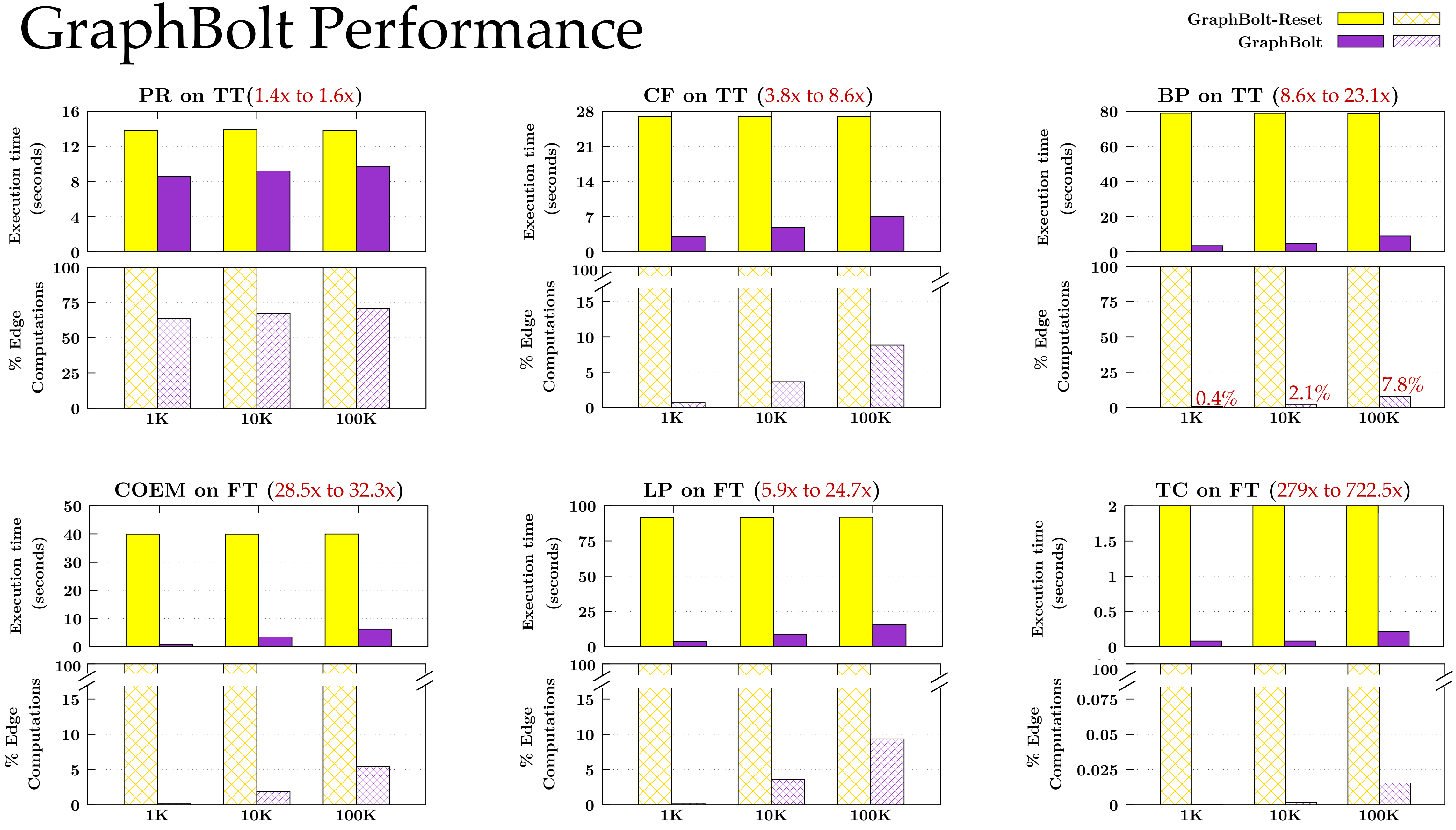
GraphBolt Performance



GraphBolt Performance



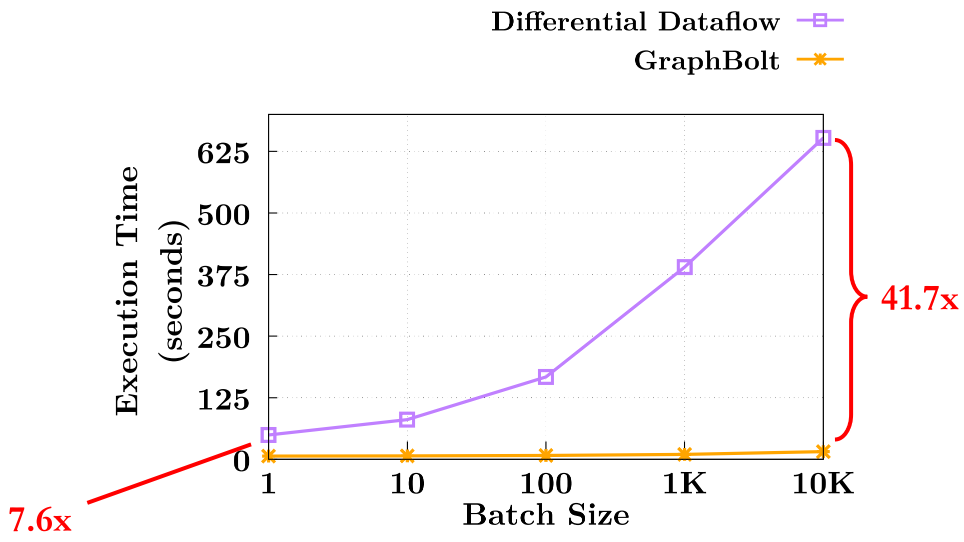
GraphBolt Performance



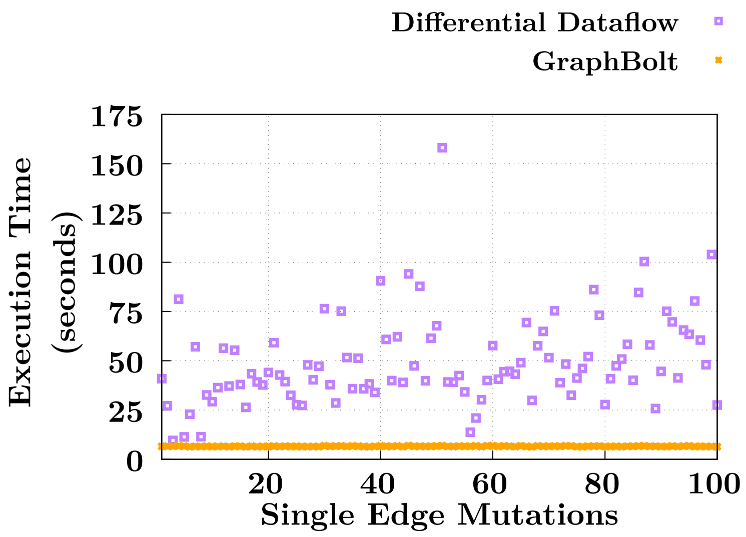
Detailed Experiments …

- Varying workload type
  - Mutations affecting majority of the values v/s affecting small values
- Varying graph mutation rate
  - Single edge update (reactiveness) v/s 1 million edge updates (throughput)
- Scaling to large graph (Yahoo) over large system (96 core/748 GB)

GraphBolt v/s Differential Dataflow



PR on TT



PR on TT
(100 single edge mutations)

Differential Dataflow. McSherry et al., CIDR 2013.

Summary

- Efficient **incremental processing** of **streaming graphs**
- Guarantee **Bulk Synchronous Parallel** semantics
- Lightweight **dependence tracking** at aggregation level
- **Dependency-aware value refinement** upon graph mutation
  - Programming model to support **incremental complex aggregation** types

Acknowledgements

SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

NSERC
CRSNG