Hyper-Flow Diffusion

Kimon Fountoulakis[1], Pan Li[2], **Shenghao Yang**[1]

[1]University of Waterloo [2]Purdue University

Hypergraph modelling are everywhere

Hypergraphs generalize graphs by allowing a hyperedge to consist of multiple nodes that capture higher-order relations in the data.

E-commerce

Nodes are products or webpages
Several products can be purchased at once
Several webpages are visited during the same session

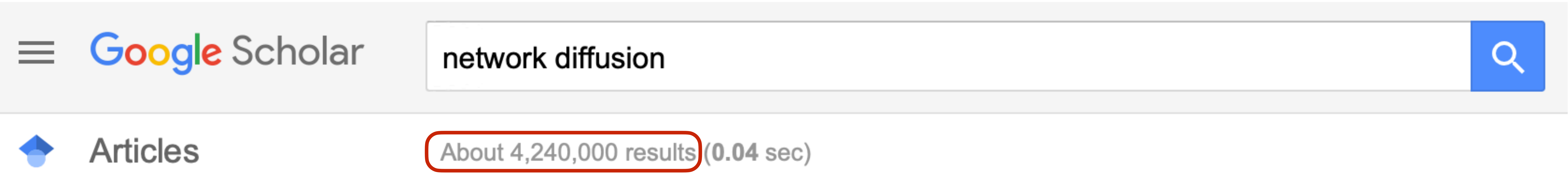
Collaboration

Nodes are authors
A group of authors collaborate on a paper/project

Ecology

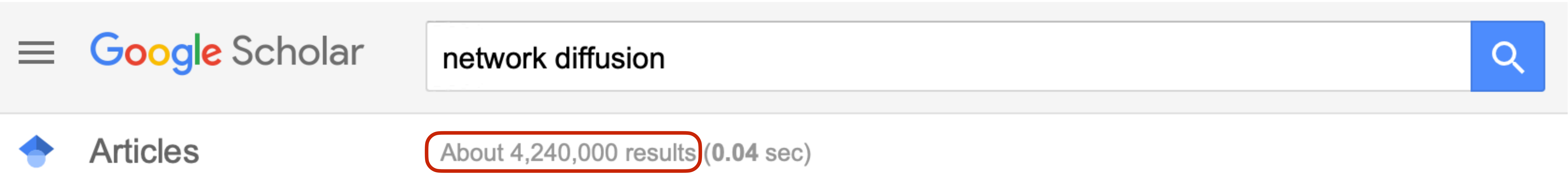
Nodes are species
Multiple species interact according to their roles in the food chain

Diffusion algorithms are everywhere (for graphs)



4.2 million results

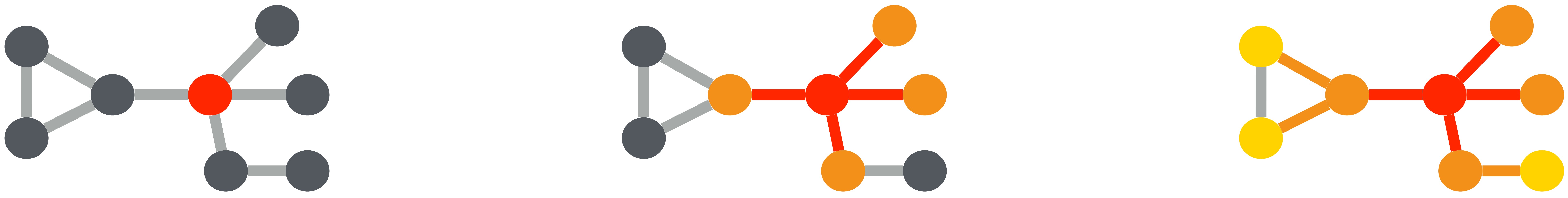
Diffusion algorithms are everywhere (for graphs)



4.2 million results

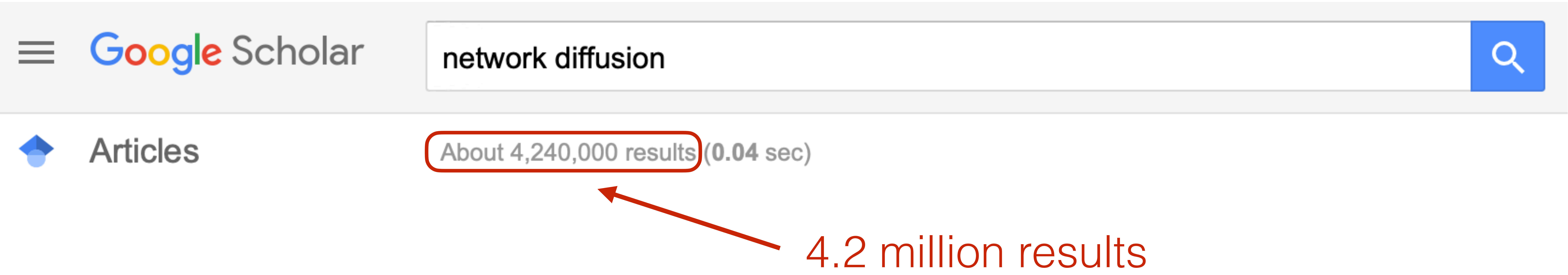
Diffusion on a graph is the process of spreading a given initial mass from some seed node(s) to neighbor nodes using the edges of the graph.

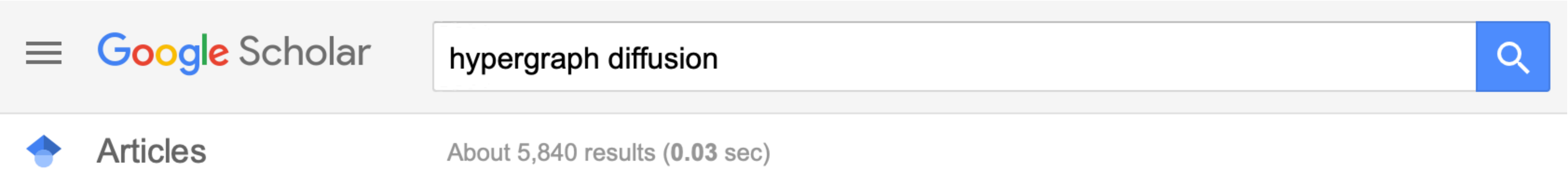
Applications include ***recommendation systems***, ***node ranking***, ***community detection***, ***social and biological network analysis***, etc.



1 ***2*** ***3***

Diffusion algorithms are everywhere (for graphs)





Hypergraph diffusion has been significantly less explored:
Existing methods either do not have a tight theoretical implication, or do not model complex high-order relations, or are not scalable.

Our motivation

We propose the first local diffusion method that

- Achieves stronger theoretical guarantees for the local hypergraph clustering problem;
- Applies to a substantially richer class of higher-order relations with only a submodularity assumption;
- Permits computational efficient algorithms.

Hypergraph diffusion has been significantly less explored:
Existing methods either do not have a **tight theoretical implication**, or do not model **complex high-order relations**, or are not **scalable**.

Our motivation

We propose the first local diffusion method that

- Achieves stronger theoretical guarantees for the local hypergraph clustering problem;
- Applies to a substantially richer class of higher-order relations with only a submodularity assumption;
- Permits computational efficient algorithms.

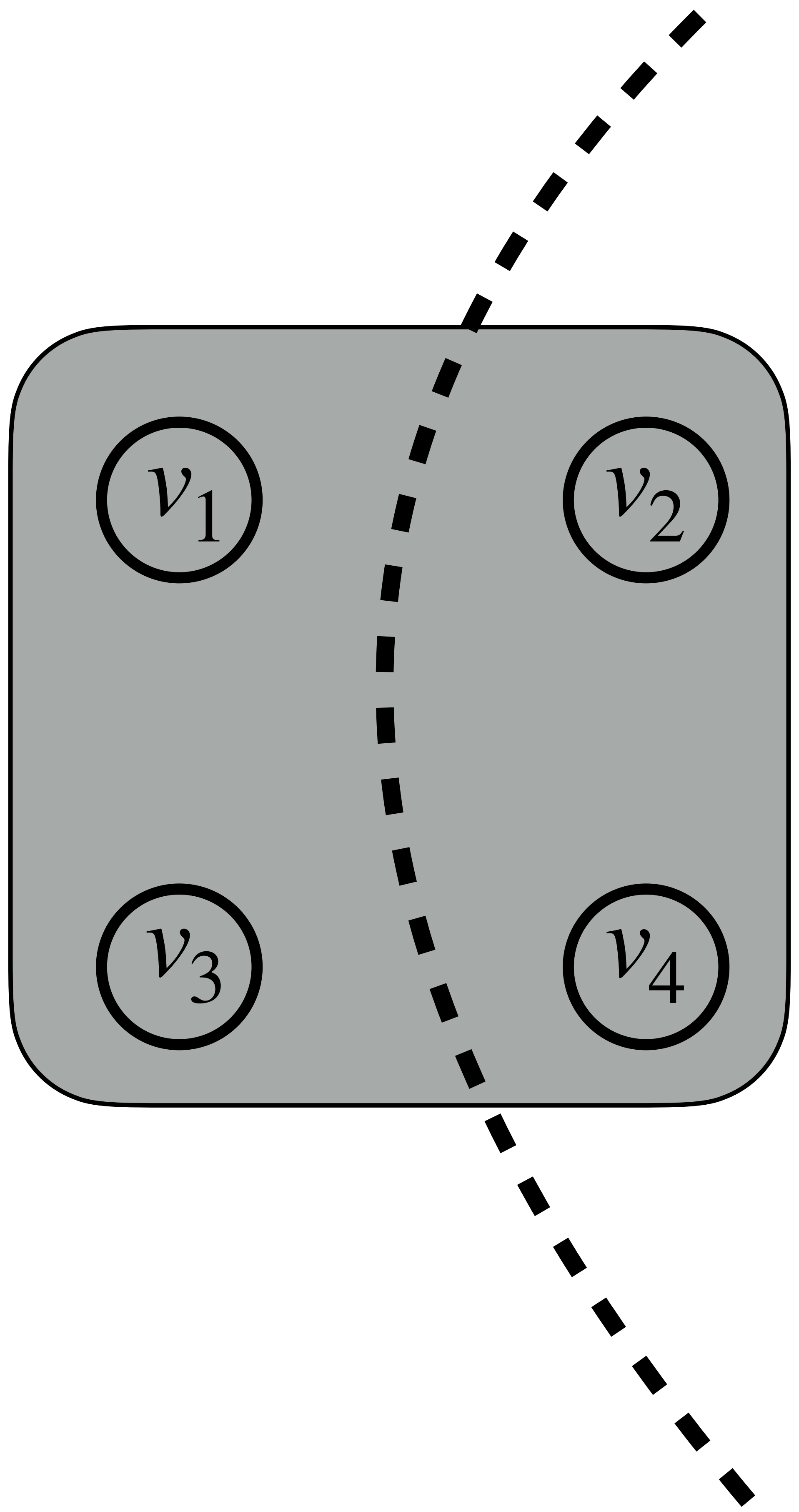
Connection to **a nonlinear hypergraph Laplacian operator**
will become clear later

Hypergraph diffusion has been significantly less explored:
Existing methods either do not have a **tight theoretical implication**, or do not model **complex high-order relations**, or are not **scalable**.

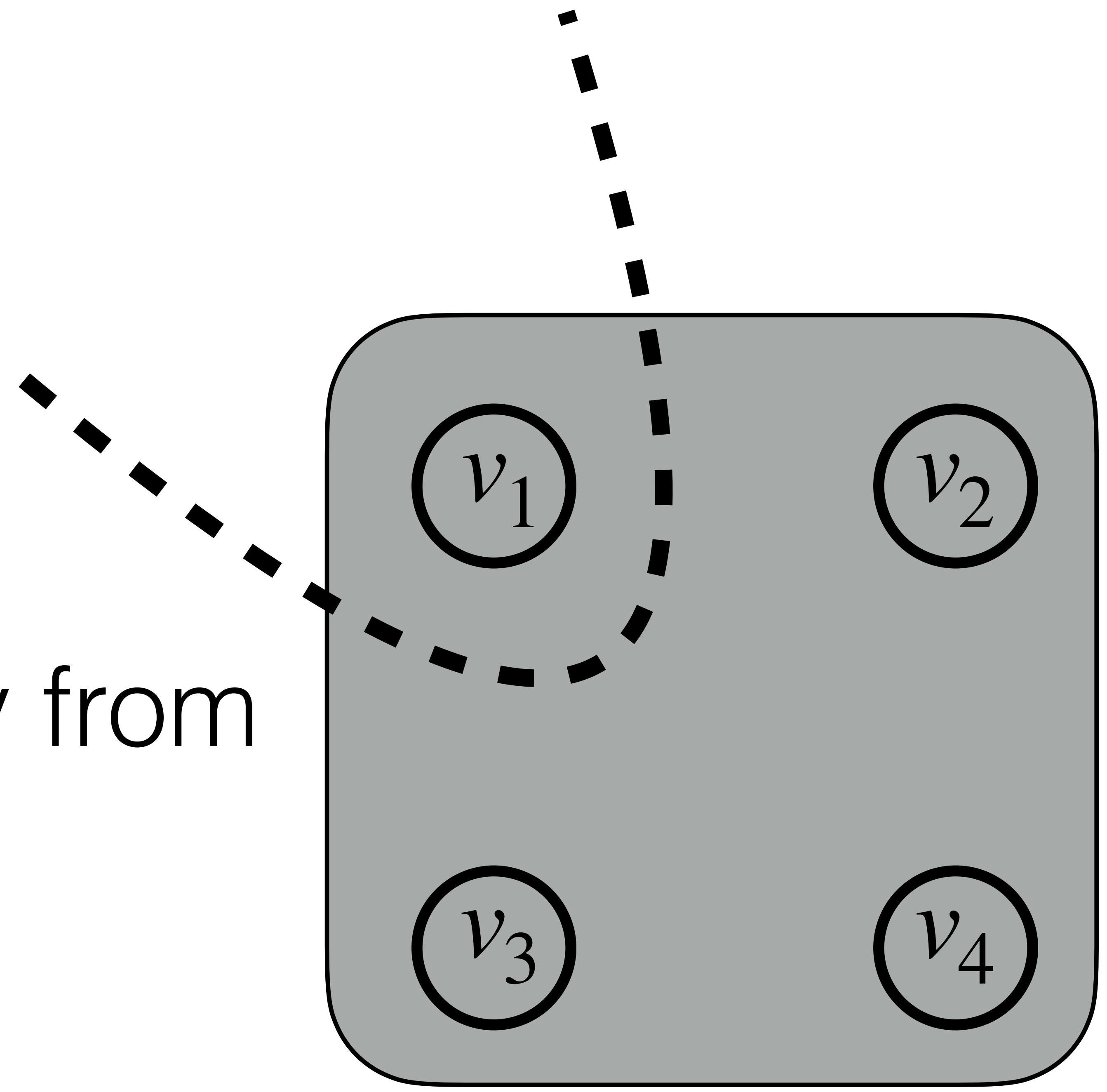
Higher-order relations: hyperedge cut perspective

There are distinct ways to cut a 4-node hyperedge.

How do we treat



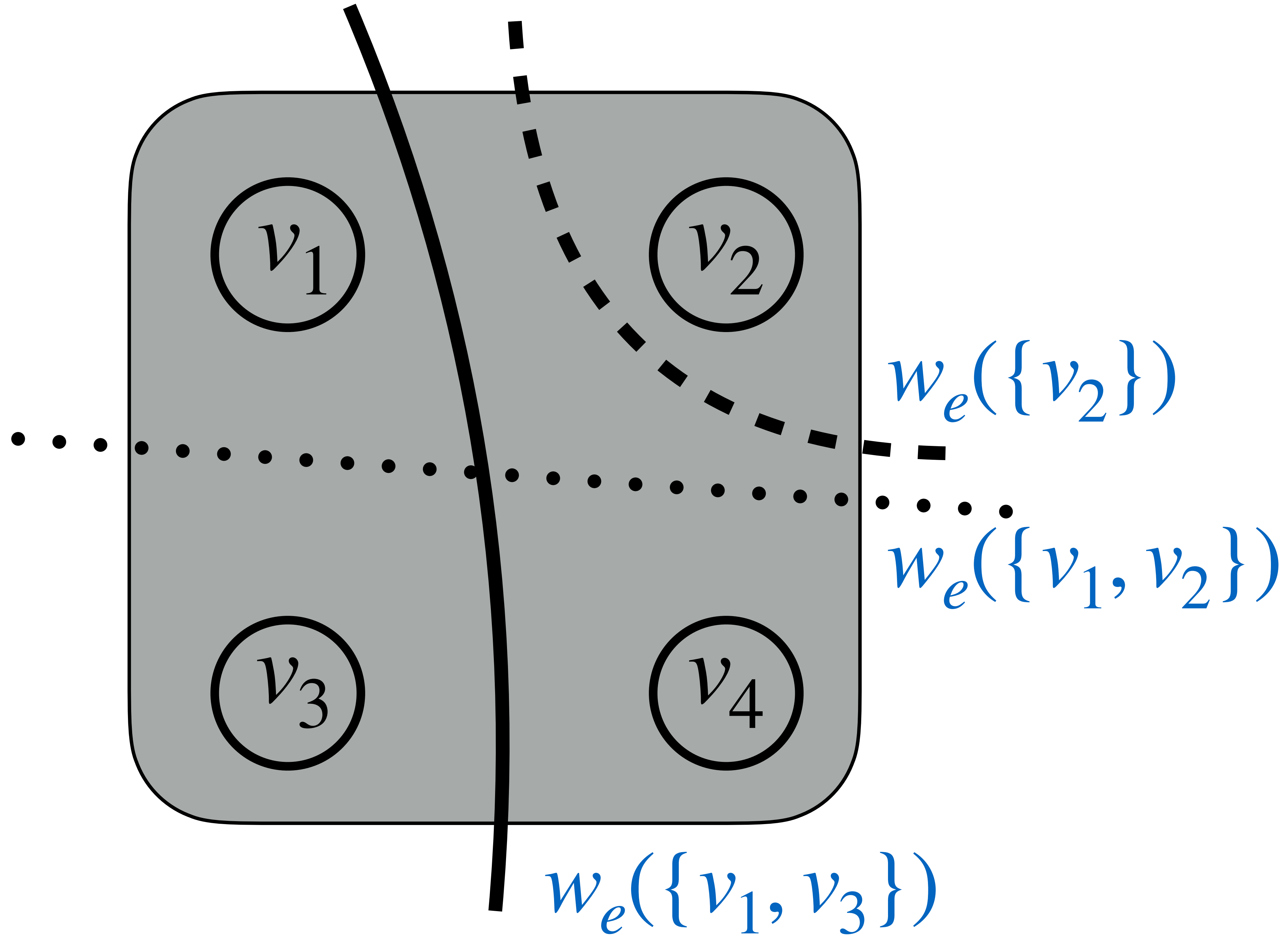
differently from



?

Higher-order relations: hyperedge cut perspective

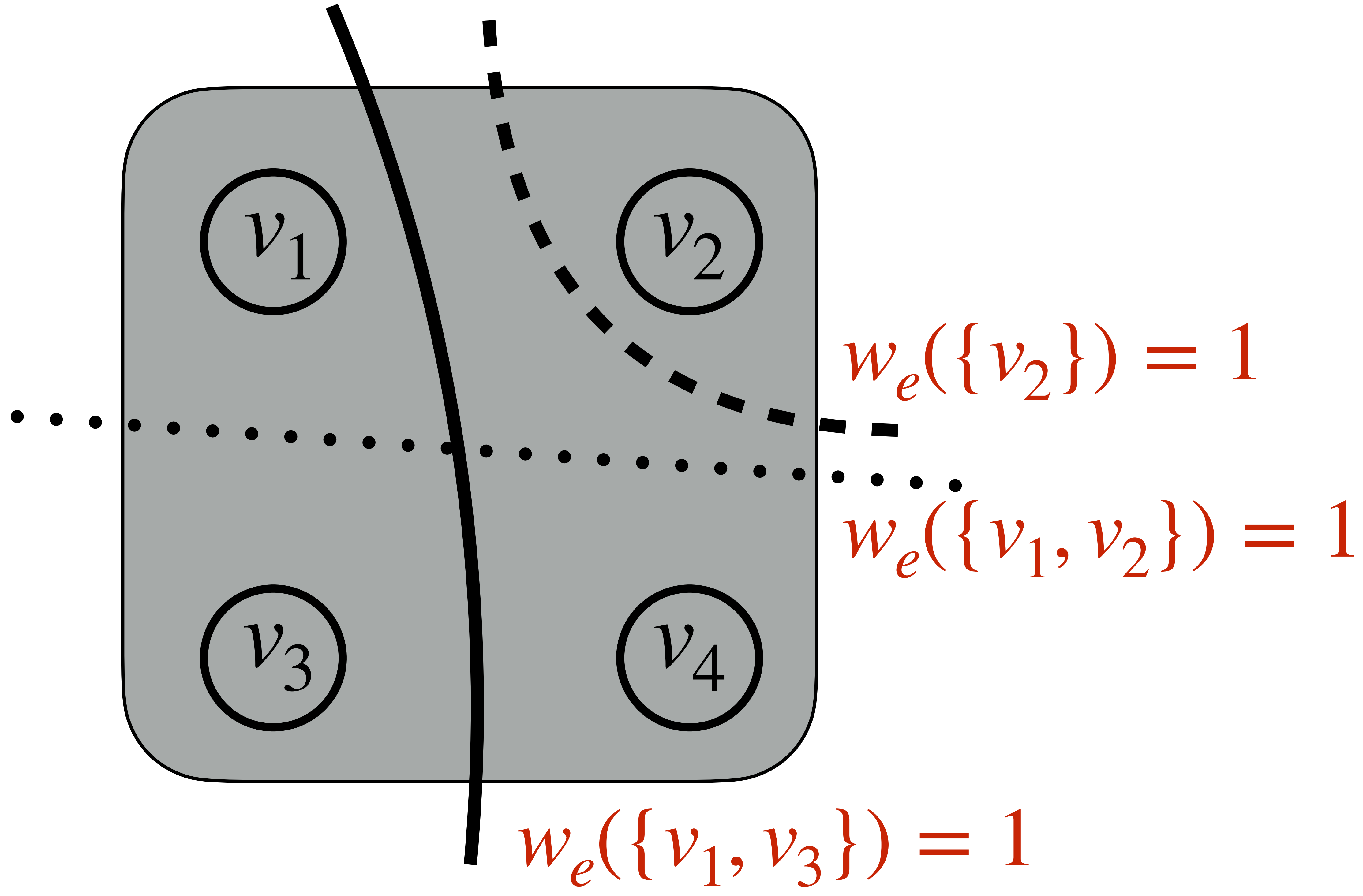
Distinct ways to cut a 4-node hyperedge may have different costs.



$w_e(S)$ specifies the cost of splitting e into S and $e \backslash S$.

Higher-order relations: hyperedge cut perspective

Distinct ways to cut a 4-node hyperedge may have different costs.

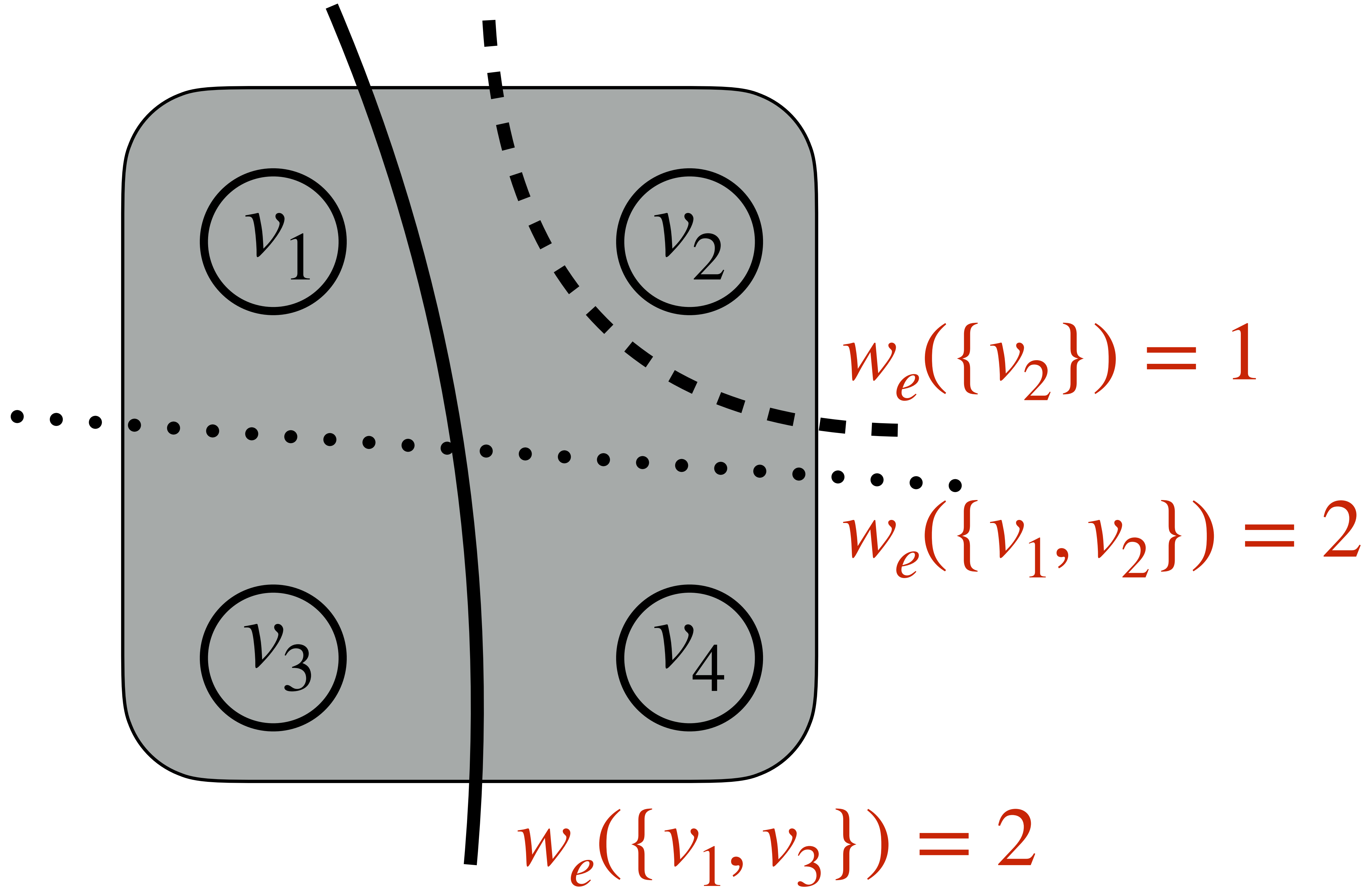


$w_e(S)$ specifies the cost of splitting e into S and $e \backslash S$.

Unit: the cost of cutting a hyperedge is always 1, i.e., $w_e(S) = 1$

Higher-order relations: hyperedge cut perspective

Distinct ways to cut a 4-node hyperedge may have different costs.



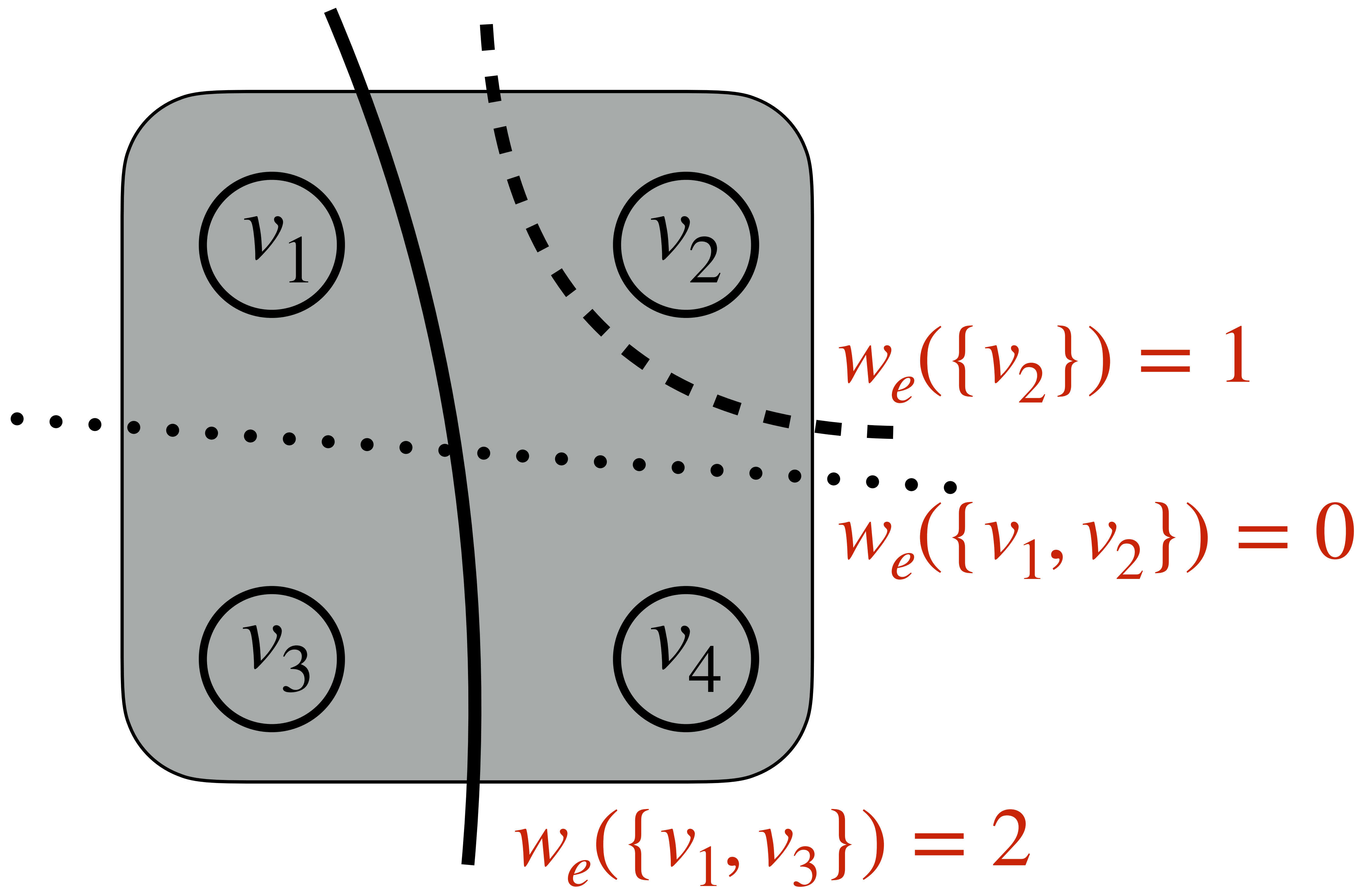
$w_e(S)$ specifies the cost of splitting e into S and $e \backslash S$.

Unit: the cost of cutting a hyperedge is always 1, i.e., $w_e(S) = 1$.

Cardinality-based: the cost of cutting a hyperedge depends on the number of nodes in either side of the hyperedge, i.e., $w_e(S) = f(\min\{|S|, |e \backslash S|\})$.

Higher-order relations: hyperedge cut perspective

Distinct ways to cut a 4-node hyperedge may have different costs.



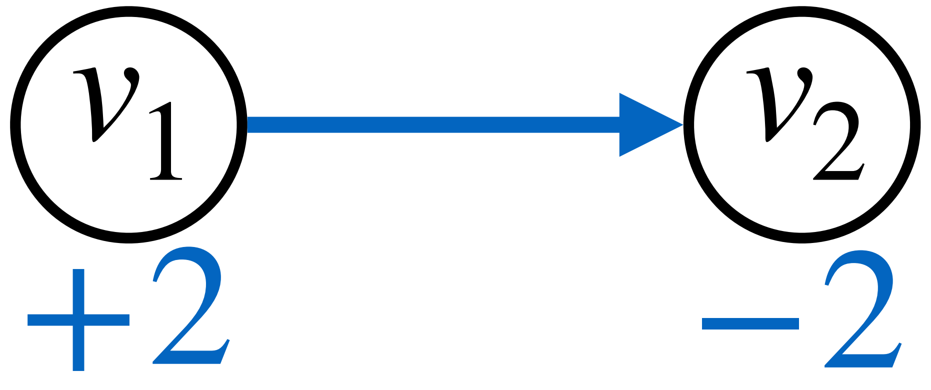
$w_e(S)$ specifies the cost of splitting e into S and $e \backslash S$.

Unit: the cost of cutting a hyperedge is always 1, i.e., $w_e(S) = 1$.

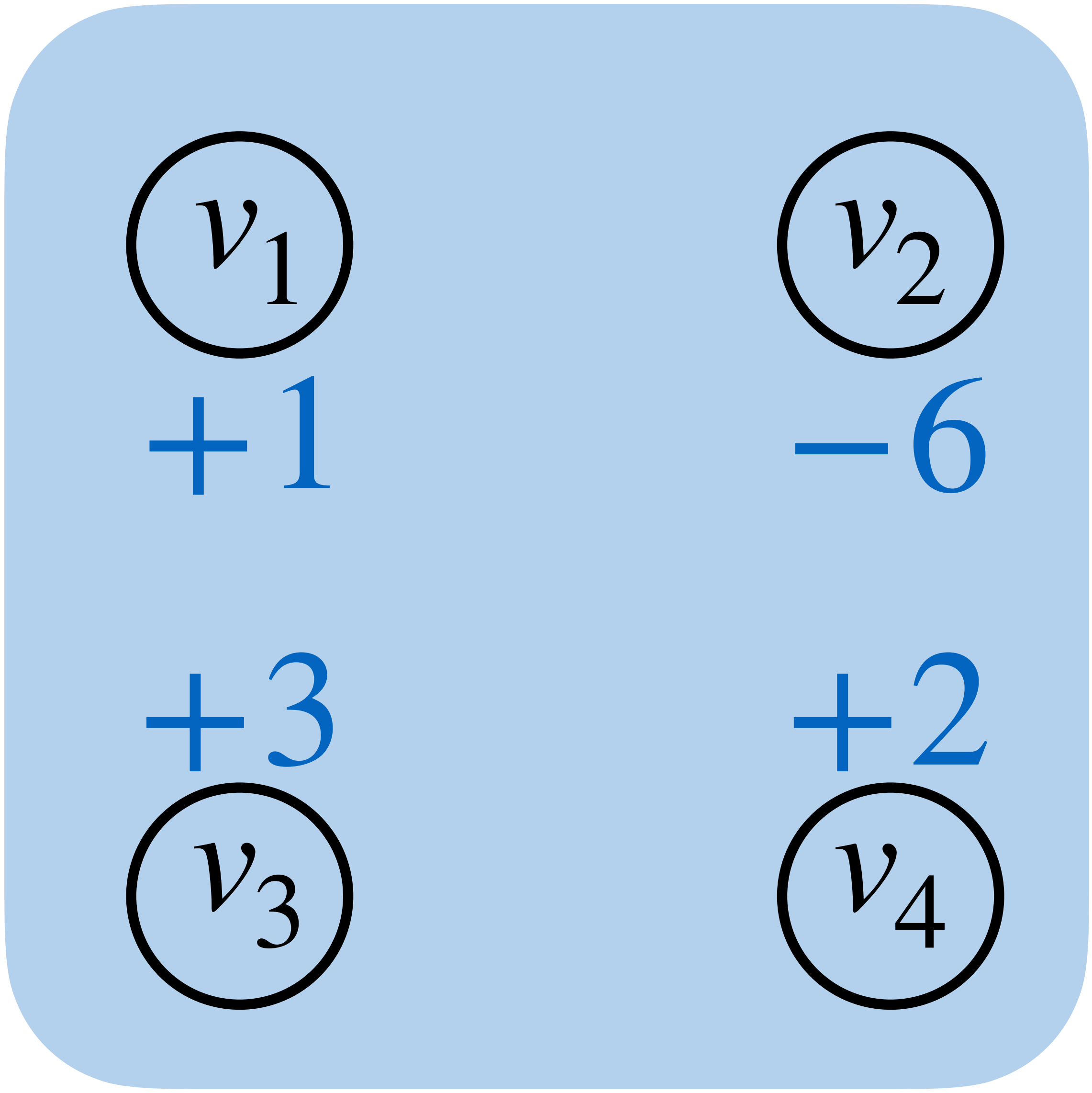
Cardinality-based: the cost of cutting a hyperedge depends on the number of nodes in either side of the hyperedge, i.e., $w_e(S) = f(\min\{|S|, |e \backslash S|\})$.

Submodular: the costs of cutting a hyperedge form a submodular function, i.e., $w_e : 2^e \to \mathbb{R}$ is a submodular set function.

Higher-order relations: hyperedge flow perspective



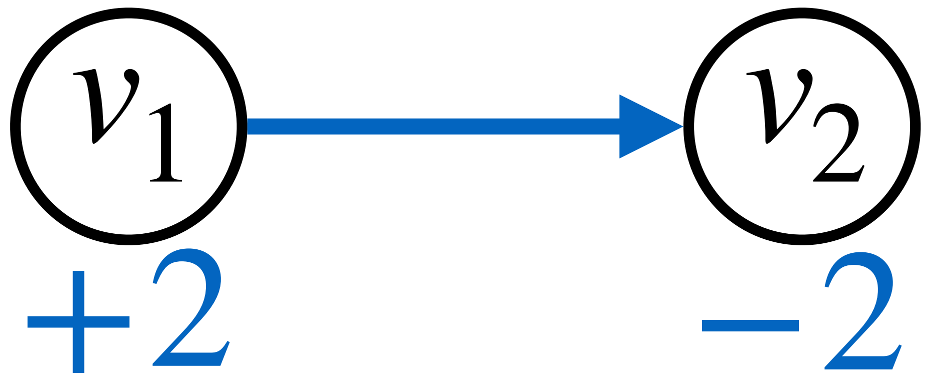
Graph edge



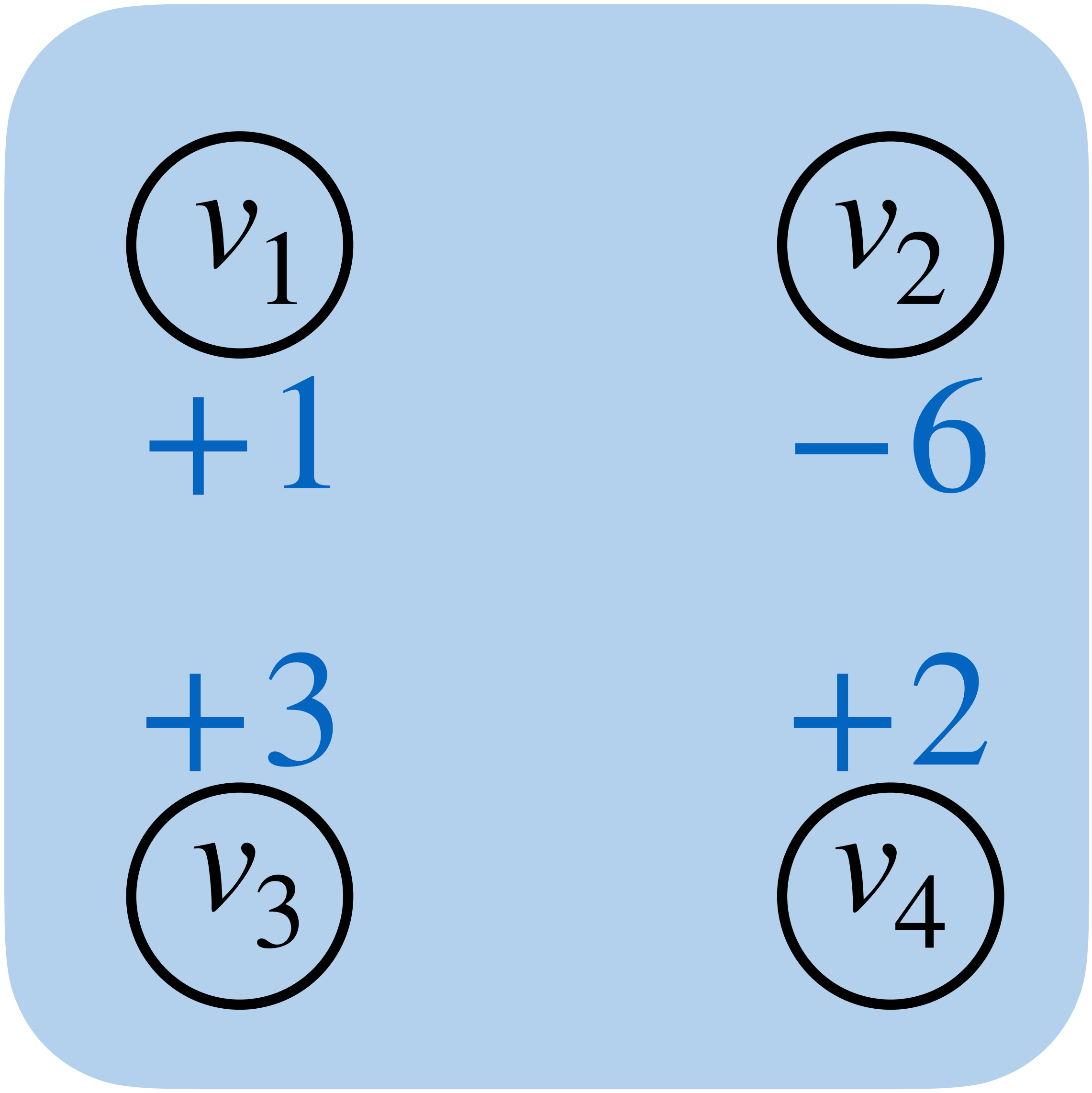
Hyperedge

For each hyperedge e, we have a vector r_e specifying the flow values.
E.g., $r_e(v_1) = 1$, $r_e(v_2) = -6$. Flow conservation: entries in r_e sums to 0.

Higher-order relations: hyperedge flow perspective



Graph edge

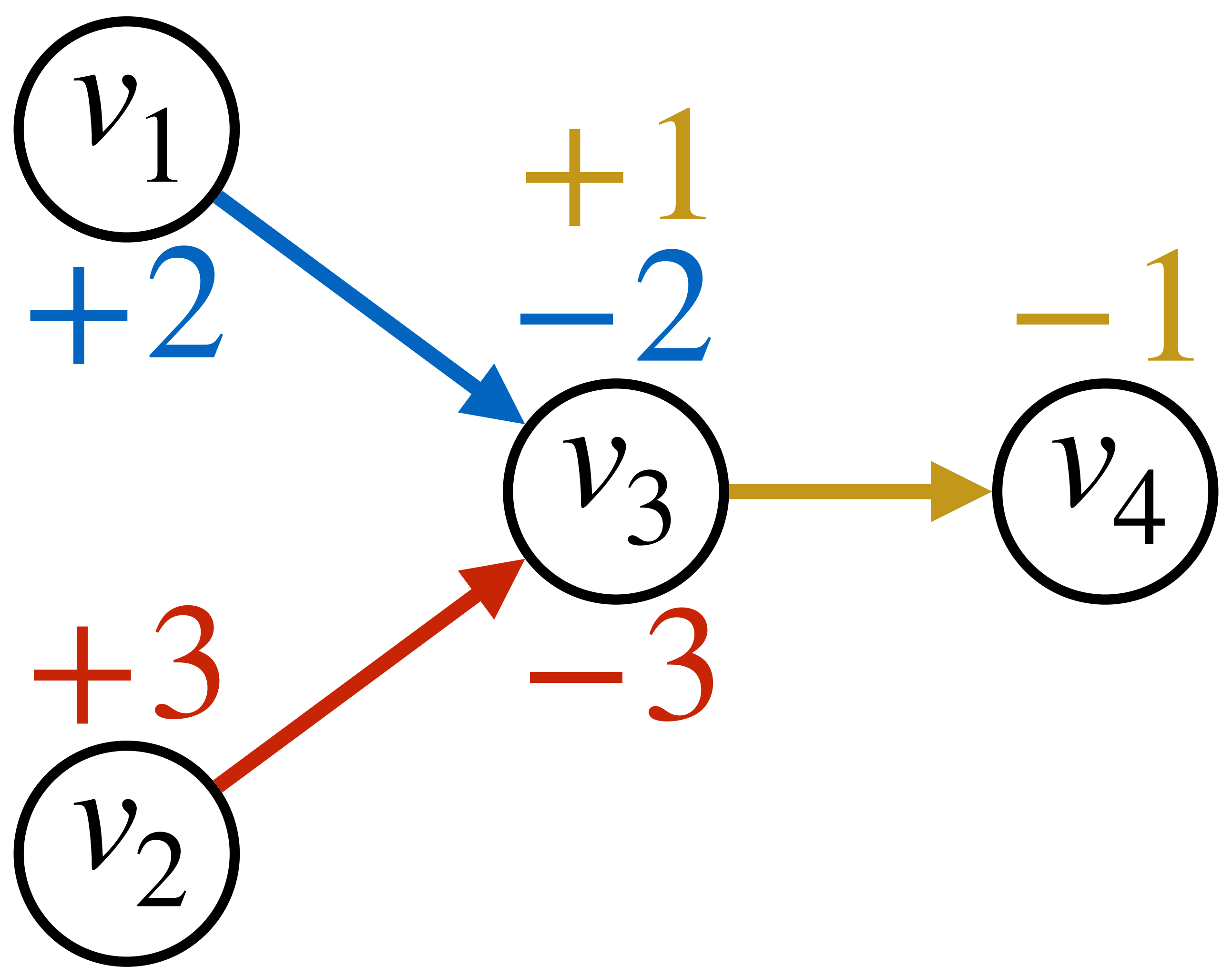


Hyperedge

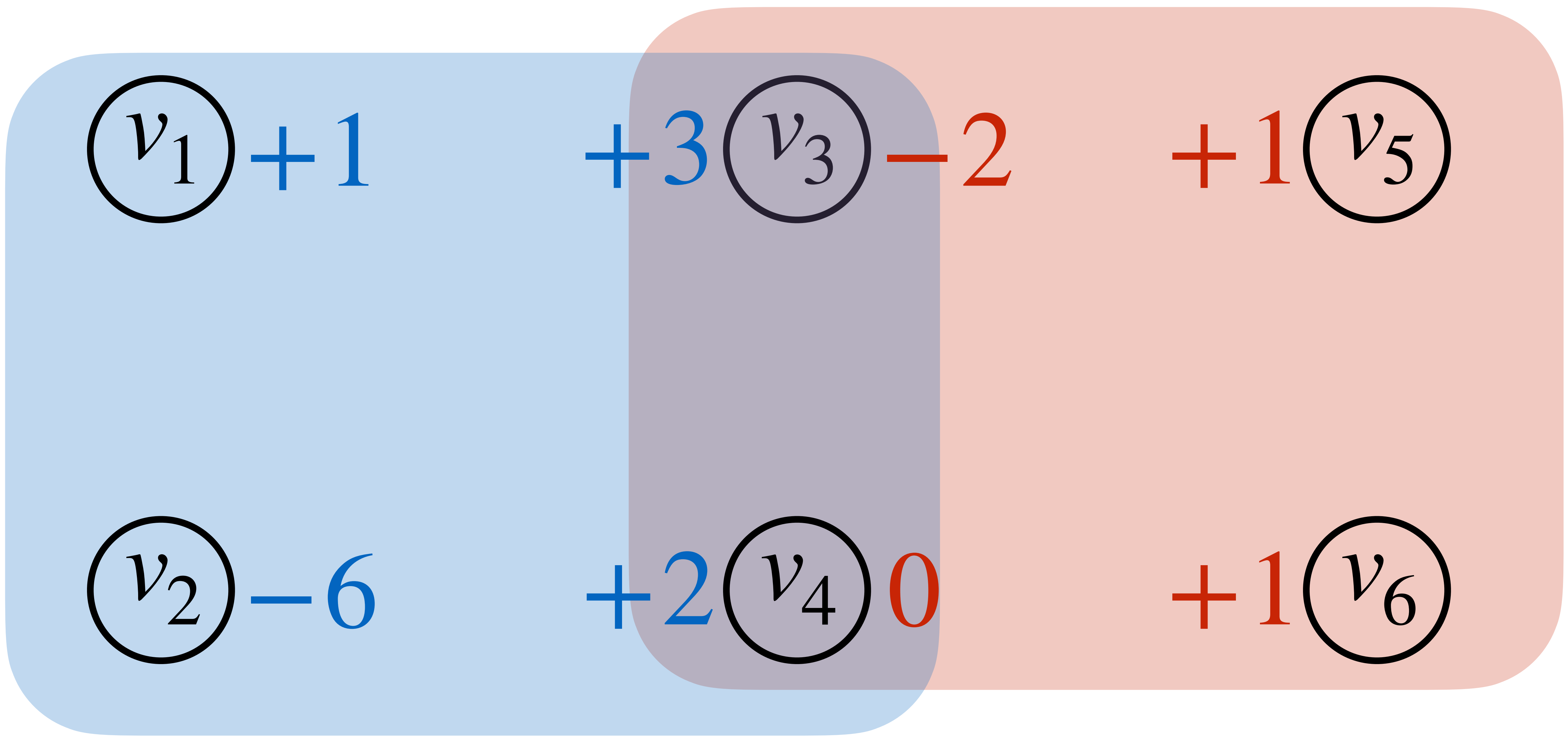
For each hyperedge e, we have a vector r_e specifying the flow values. E.g., $r_e(v_1) = 1$, $r_e(v_2) = -6$. Flow conservation: entries in r_e sums to 0.

Additional constraints on r_e can make the flow values respect higher-order relations.

Higher-order relations: hyperedge flow perspective



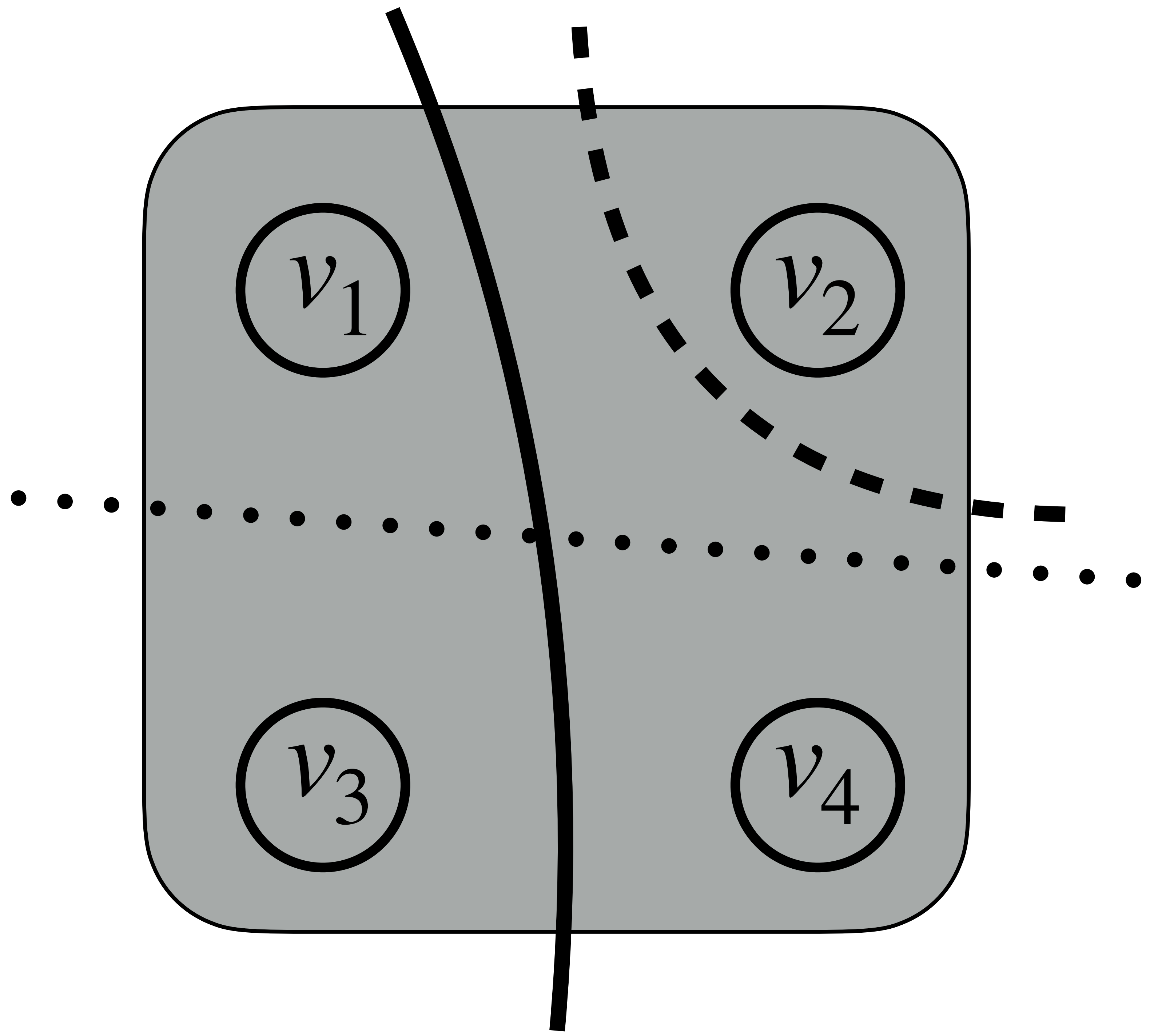
Flows on graph

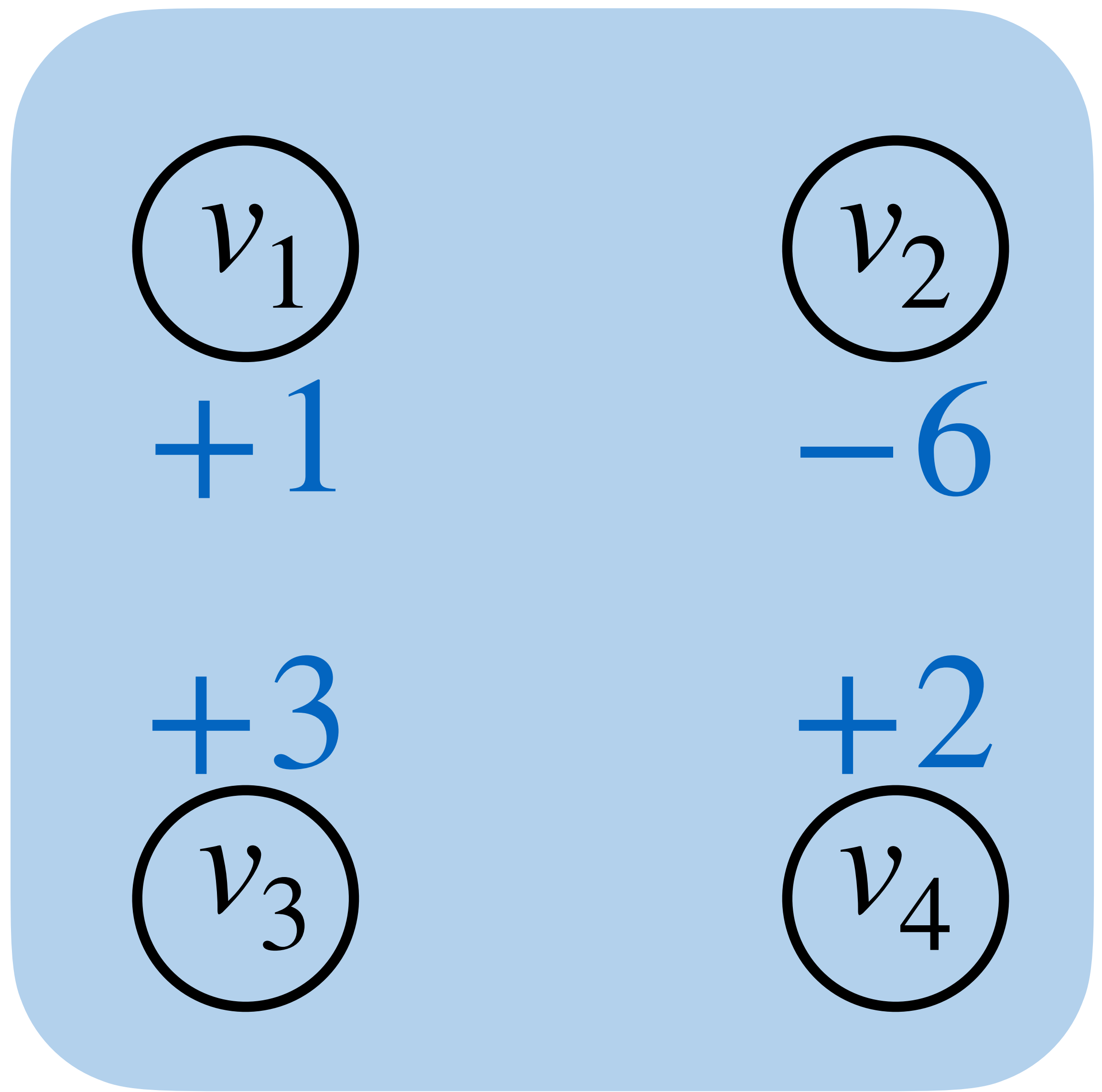


Flows on hypergraph

A natural generalization of network flows.

Higher-order relations: primal-dual flow/cut connection





- w_e is a set function on e
- $w_e(S)$ specifies the **cut-cost** of splitting e into S and $e \backslash S$
- w_e is submodular

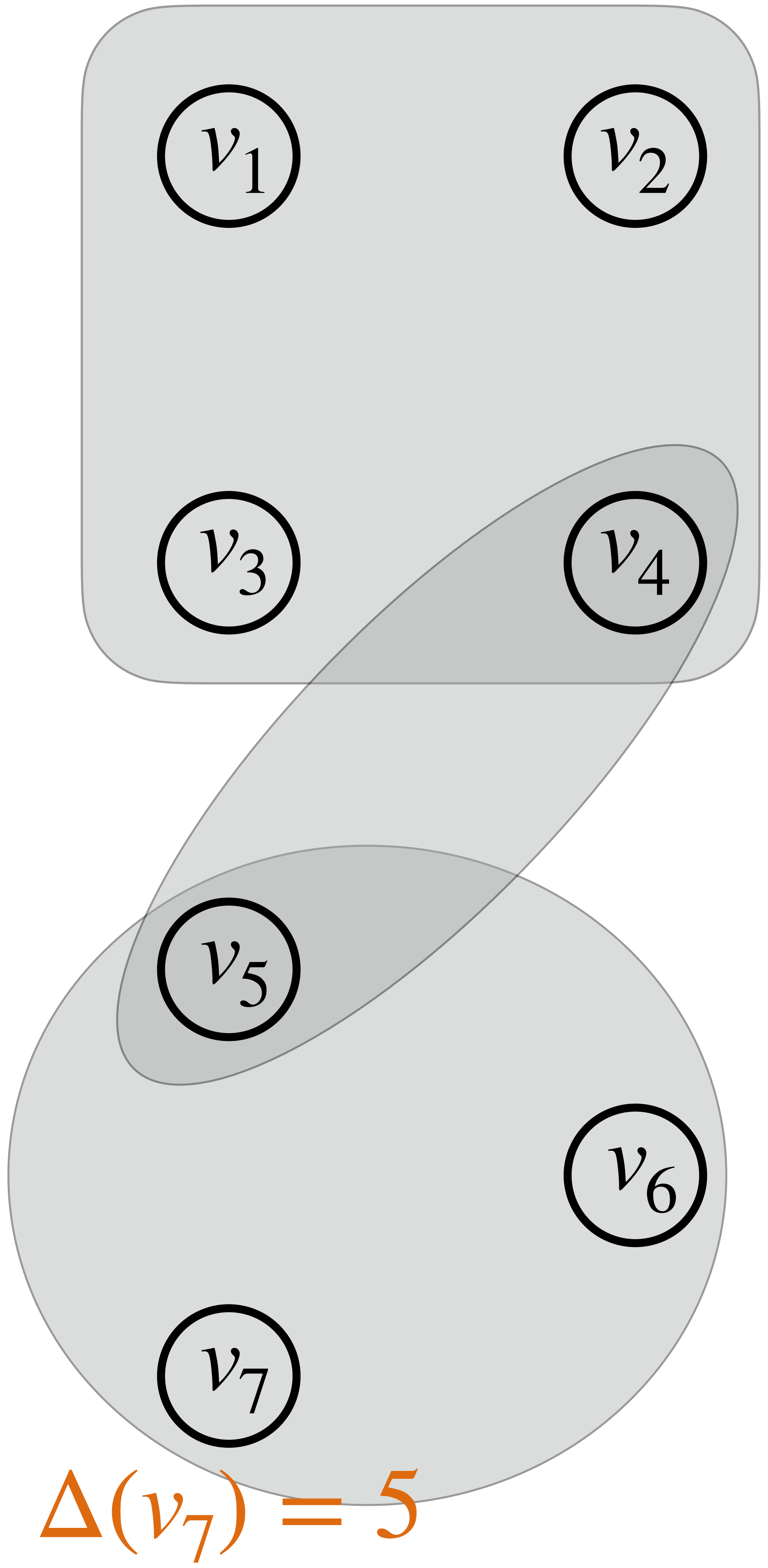
- r_e is a vector in $\mathbb{R}^{|e|}$
- r_e specifies the **flow** over e
- r_e lies in $\mathbb{R}_+(B_e)$

Cone generated by the base polytope of w_e

Hyper-flow diffusion: definition and notation

Consider a hypergraph $H = (V, E)$

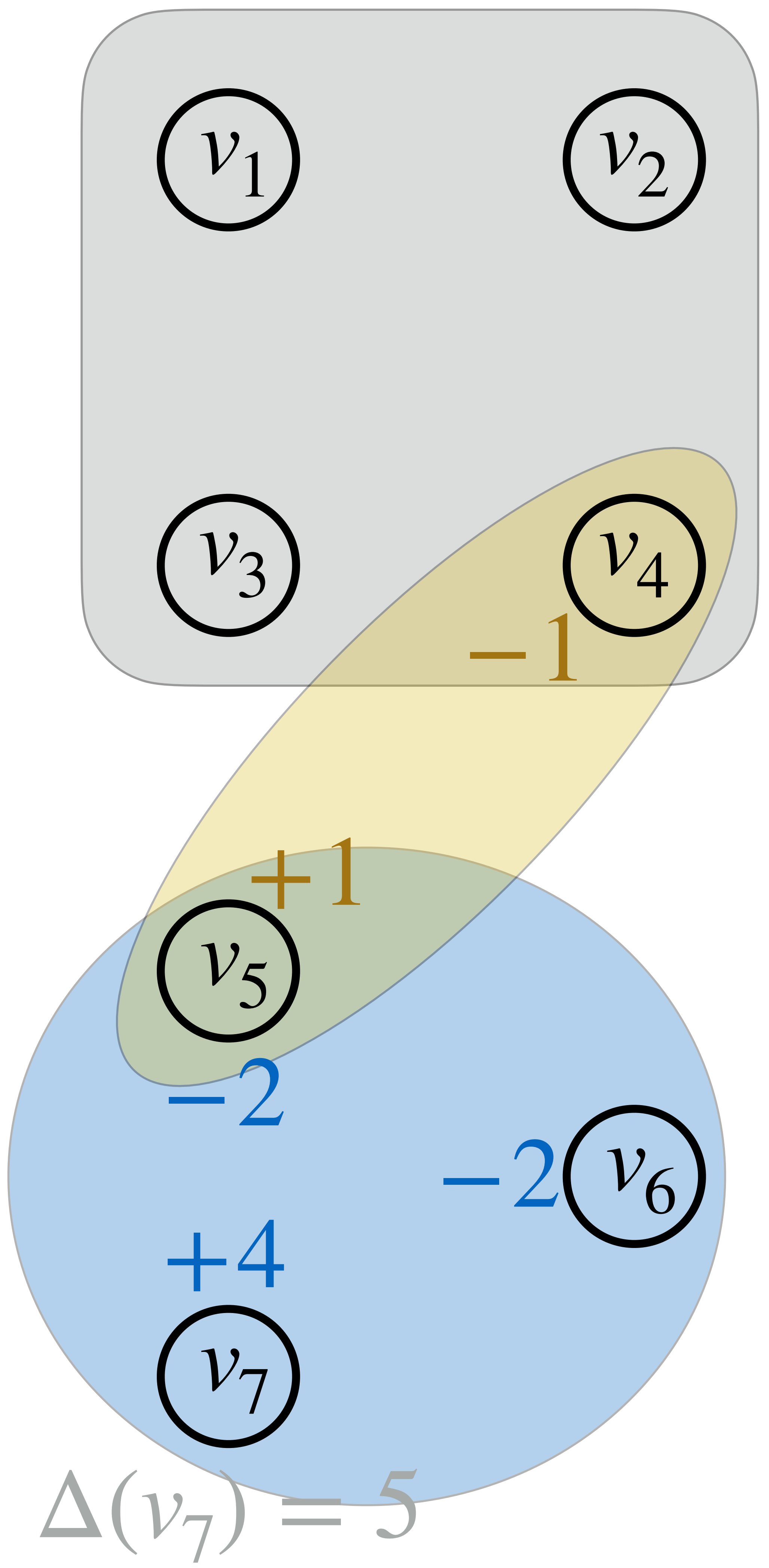
- $\Delta \in \mathbb{R}_+^{|V|}$ specifies **initial mass** on nodes.



Hyper-flow diffusion: definition and notation

Consider a hypergraph $H = (V, E)$

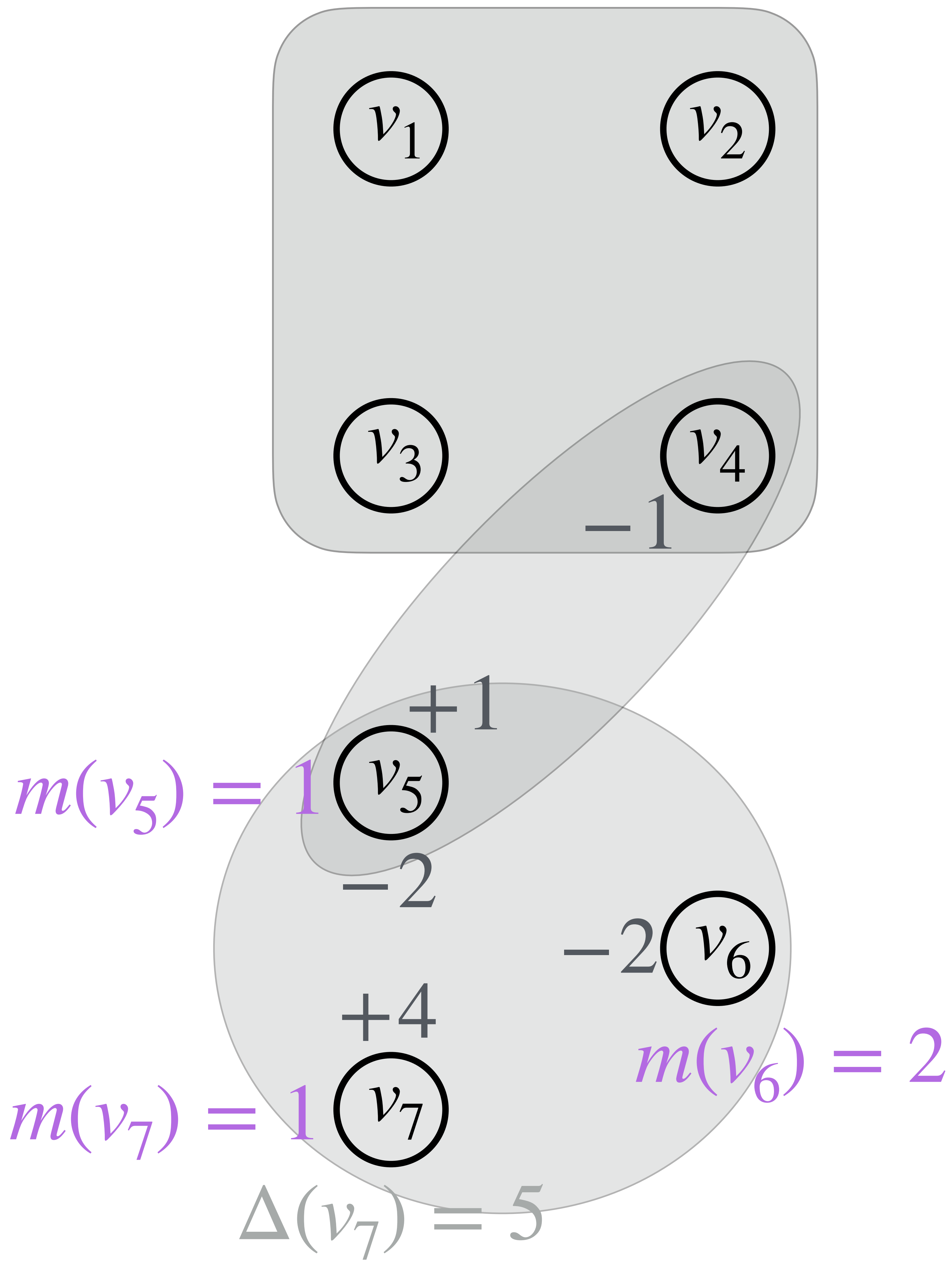
- $\Delta \in \mathbb{R}_+^{|V|}$ specifies initial mass on nodes
- r_e, $e \in E$, specifies the **flow routings**



Hyper-flow diffusion: definition and notation

Consider a hypergraph $H = (V, E)$

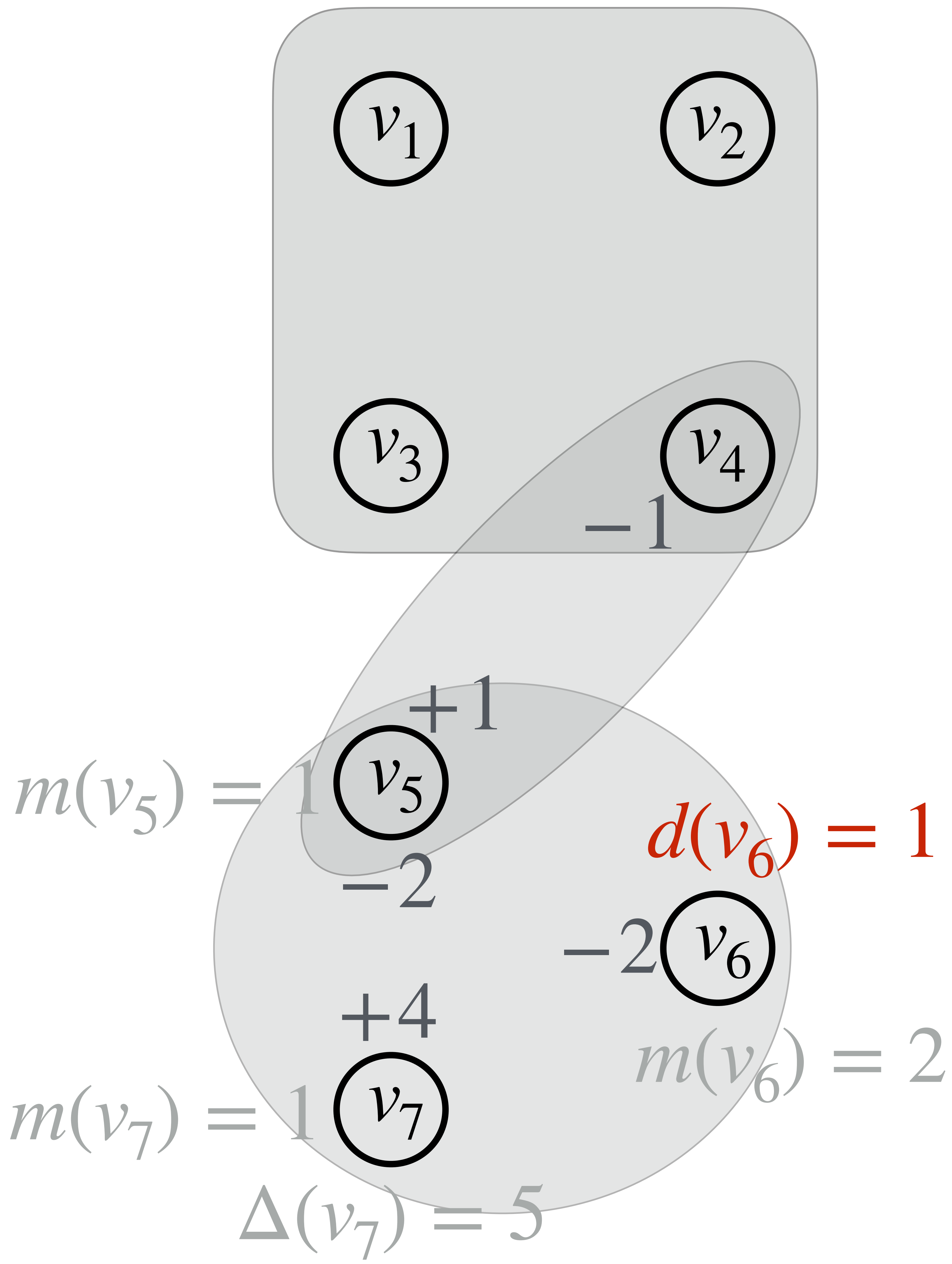
- $\Delta \in \mathbb{R}_+^{|V|}$ specifies initial mass on nodes
- r_e, $e \in E$, specifies the flow routings
- $m := \Delta - \sum_{e \in E} r_e$ specifies **net mass** on nodes



Hyper-flow diffusion: definition and notation

Consider a hypergraph $H = (V, E)$

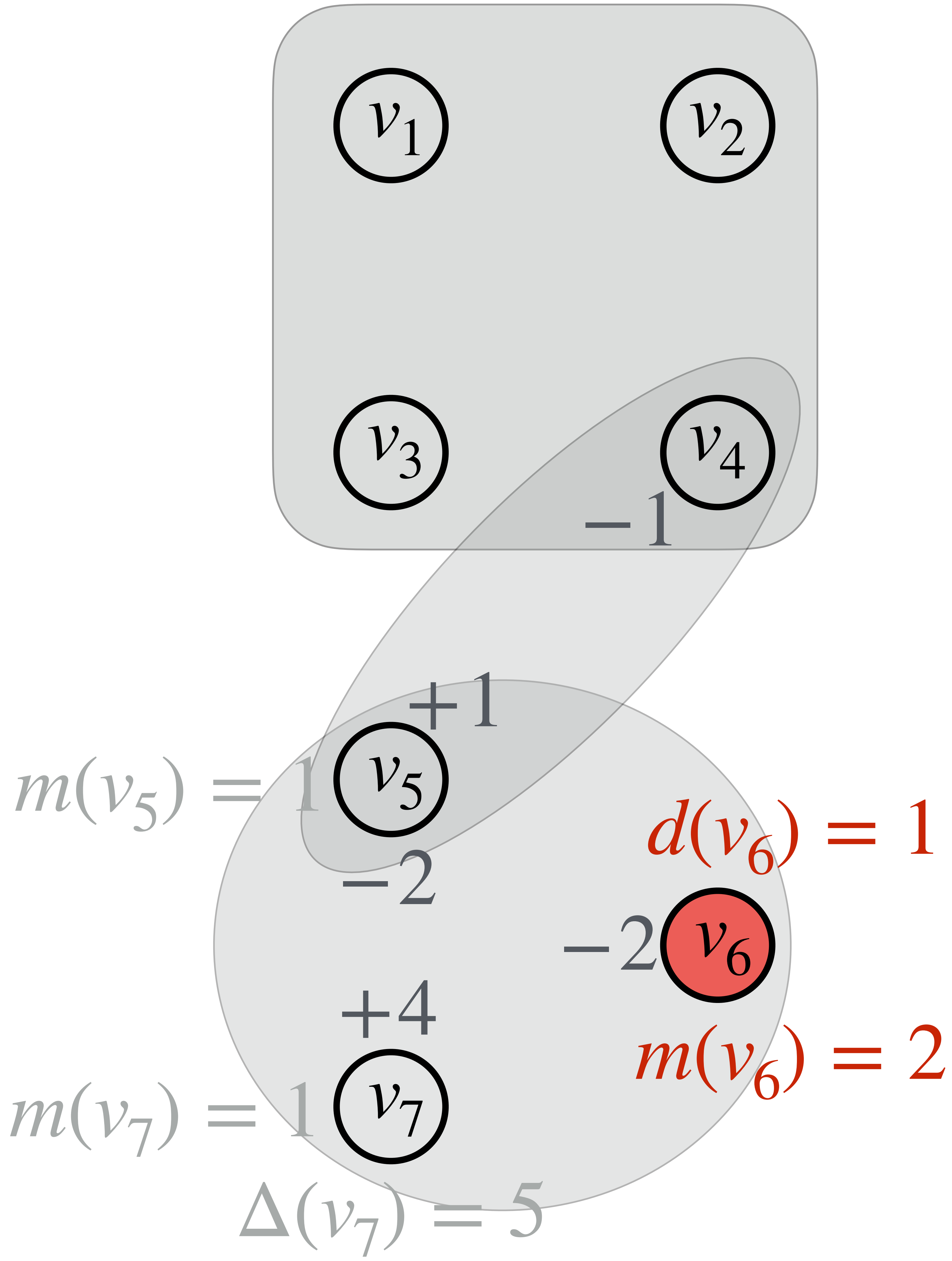
- $\Delta \in \mathbb{R}_+^{|V|}$ specifies initial mass on nodes
- r_e, $e \in E$, specifies the flow routings
- $m := \Delta - \sum_{e \in E} r_e$ specifies net mass on nodes
- Each node has **capacity** equal to its degree



Hyper-flow diffusion: definition and notation

Consider a hypergraph $H = (V, E)$

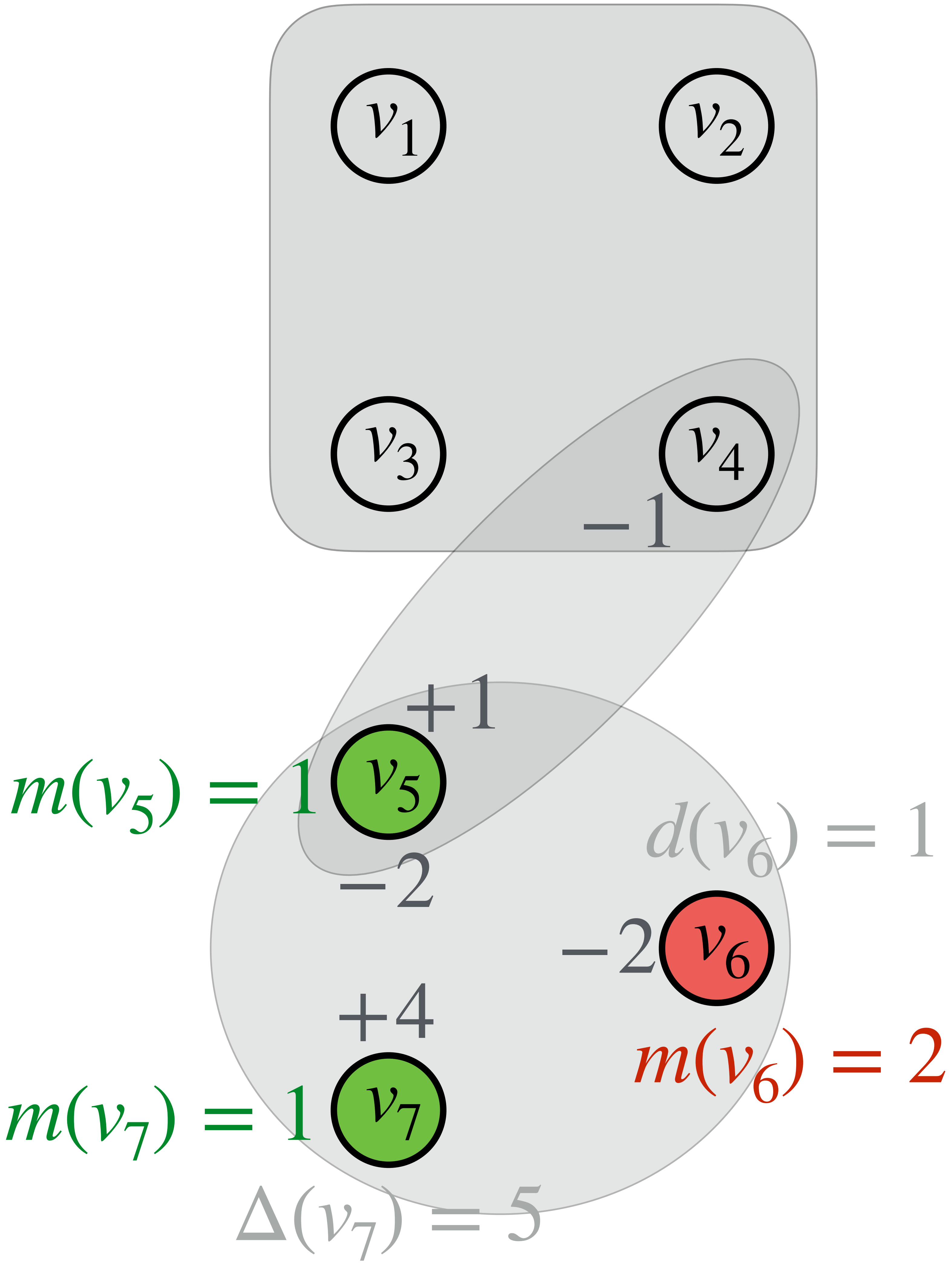
- $\Delta \in \mathbb{R}_+^{|V|}$ specifies initial mass on nodes
- $r_e, e \in E$, specifies the flow routings
- $m := \Delta - \sum_{e \in E} r_e$ specifies net mass on nodes
- Each node has **capacity** equal to its degree



Hyper-flow diffusion: definition and notation

Consider a hypergraph $H = (V, E)$

- $\Delta \in \mathbb{R}_+^{|V|}$ specifies initial mass on nodes
- r_e, $e \in E$, specifies the flow routings
- $m := \Delta - \sum_{e \in E} r_e$ specifies net mass on nodes
- Each node has capacity equal to its degree
- A set of flow routings r_e, $e \in E$, is **feasible** if $m(v) \leq d(v), \forall v$



Hyper-flow diffusion: formulations

Given $H = (V, E)$, cut-costs w_e for $e \in E$, initial mass Δ, our diffusion problem finds **feasible** flow routings with **minimum ℓ_2-norm** cost.

$$\min \frac{1}{2} \sum_{e \in E} \phi_e^2$$ ← ϕ_e is magnitude of flow (discussed later)

$$m(v) \leq d(v), \forall v$$ ← Capacity constraint forces diffusion of initial mass

$$\sum_{v \in e} r_e(v) = 0, \forall e$$ ← Flow conservation on a hyperedge

Hyper-flow diffusion: formulations

Given $H = (V, E)$, cut-costs w_e for $e \in E$, initial mass Δ, our diffusion problem finds **feasible** flow routings with **minimum ℓ_2-norm** cost.

$$\min \frac{1}{2} \sum_{e \in E} \phi_e^2$$ ← ϕ_e is magnitude of flow (discussed later)

$$m(v) \leq d(v), \forall v$$ ← Capacity constraint forces diffusion of initial mass

~~$$\sum_{v \in e} r_e(v) = 0, \forall e$$~~ Flow conservation does not model nontrivial higher-order relations

$$r_e \in \phi_e B_e, \forall e$$ ← New constraint that reflects higher-order relations

Hyper-flow diffusion: formulations

Given $H = (V, E)$, cut-costs w_e for $e \in E$, initial mass Δ, our diffusion problem finds **feasible** flow routings with **minimum ℓ_2-norm** cost.

$$\min \frac{1}{2} \sum_{e \in E} \phi_e^2$$ ← ϕ_e is magnitude of flow

$$m(v) \le d(v), \forall v$$ ← Capacity constraint forces diffusion of initial mass

~~$\sum_{v \in e} r_e(v) = 0, \forall e$~~ Flow conservation does not model nontrivial higher-order relations

$$r_e \in \phi_e B_e, \forall e$$ ← New constraint that reflects higher-order relations

Magnitude of flow (ϕ_e)

$$B_e = \{\rho_e \in \mathbb{R}^{|V|} : \rho_e(S) \le w_e(S) \forall S \subseteq V, \rho_e(V) = w_e(V)\}$$

The base polytope for w_e

Hyper-flow diffusion: formulations

Given $H = (V, E)$, cut-costs w_e for $e \in E$, initial mass Δ, our diffusion problem finds **feasible** flow routings with **minimum ℓ_2-norm** cost.

$$\min \frac{1}{2} \sum_{e \in E} \phi_e^2$$ ⟵ ϕ_e is magnitude of flow

$$m(v) \leq d(v), \forall v$$ ⟵ Capacity constraint forces diffusion of initial mass

$$r_e \in \phi_e B_e, \forall e$$ ⟵ Flow constraint encodes high-order relations

Hyper-flow diffusion: formulations

Given $H = (V, E)$, cut-costs w_e for $e \in E$, initial mass Δ, our diffusion problem finds **feasible** flow routings with **minimum ℓ_2-norm** cost.

$$\min \ \frac{1}{2}\sum_{e\in E}\phi_e^2 + \frac{\sigma}{2}\sum_{v\in V} d(v)z_v^2$$

$$m(v) \le d(v) + \sigma d(v) z_v, \forall v$$

$$r_e \in \phi_e B_e, \forall e$$

For computational efficiency reasons we introduce a hyper-parameter $\sigma \ge 0$

Hyper-flow diffusion: formulations

Given $H = (V, E)$, cut-costs w_e for $e \in E$, initial mass Δ, our diffusion problem finds **feasible** flow routings with **minimum ℓ_2-norm** cost.

$$\min \ \frac{1}{2}\sum_{e\in E}\phi_e^2 + \frac{\sigma}{2}\sum_{v\in V} d(v) z_v^2$$

$$m(v) \le d(v) + \sigma d(v) z_v, \forall v$$

$$r_e \in \phi_e B_e, \forall e$$

For computational efficiency reasons we introduce a hyper-parameter $\sigma \ge 0$

The dual problem is $$\min_{x\ge 0} \ \frac{1}{2}\sum_{e\in E} f_e(x)^2 + \frac{\sigma}{2}\sum_{v\in V} d(v) x_v^2 + (d-\Delta)^T x$$

Quadratic form w.r.t. **Nonlinear hypergraph Laplacian operator**
Reduces to $x^T L x$ for standard graphs

$f_e(x) := \max_{\rho_e \in B_e} \rho_e^T x$ is the Lovasz extension of w_e

Hyper-flow diffusion: formulations

Given $H = (V, E)$, cut-costs w_e for $e \in E$, initial mass Δ, our diffusion problem finds **feasible** flow routings with **minimum ℓ_2-norm** cost.

$$\min \frac{1}{2}\sum_{e \in E} \phi_e^2 + \frac{\sigma}{2}\sum_{v \in V} d(v) z_v^2$$

$$m(v) \leq d(v) + \sigma d(v) z_v, \forall v$$

$$r_e \in \phi_e B_e, \forall e$$

For computational efficiency reasons we introduce a hyper-parameter $\sigma \geq 0$

The dual problem is $$\min_{x \geq 0} \frac{1}{2}\sum_{e \in E} f_e(x)^2 + \frac{\sigma}{2}\sum_{v \in V} d(v) x_v^2 + (d - \Delta)^T x$$

We use the dual solution x for node ranking and clustering

Hyper-flow diffusion: local clustering guarantee

Conductance of target cluster C

$$\Phi(C) = \frac{\sum_{e\in E} w_e(C)}{\min\{\mathbf{vol}(C), \mathbf{vol}(V\backslash C)\}}$$ where $\mathbf{vol}(C) := \sum_{v\in C} d(v)$

Seed set $S := \mathbf{supp}(\Delta)$.

Assumption 1 (sufficient overlap): $\mathbf{vol}(S\cap C) \geq \beta\mathbf{vol}(S)$, $\mathbf{vol}(S\cap C) \geq \alpha\mathbf{vol}(C)$ $\quad \alpha, \beta \geq \frac{1}{\log^t \mathbf{vol}(C)}$ for some t

Assumption 2: $0 \leq \sigma \leq \beta\Phi(C)/3$

The output cluster $\tilde{C}$ satisfies $\Phi(\tilde{C}) \leq \tilde{\mathcal{O}}(\sqrt{\Phi(C)})$

Hyper-flow diffusion: local clustering guarantee

Conductance of target cluster C

$$\Phi(C) = \frac{\sum_{e \in E} w_e(C)}{\min\{\mathbf{vol}(C), \mathbf{vol}(V \backslash C)\}}$$ where $\mathbf{vol}(C) := \sum_{v \in C} d(v)$

Seed set $S := \mathbf{supp}(\Delta)$.

Assumption 1 (sufficient overlap): $\mathbf{vol}(S \cap C) \geq \beta \mathbf{vol}(S)$, $\mathbf{vol}(S \cap C) \geq \alpha \mathbf{vol}(C)$ $\quad \alpha, \beta \geq \frac{1}{\log^t \mathbf{vol}(C)}$ for some t

Assumption 2: $0 \leq \sigma \leq \beta \Phi(C)/3$

The output cluster $\tilde{C}$ satisfies $\Phi(\tilde{C}) \leq \tilde{\mathcal{O}}(\sqrt{\Phi(C)})$

The first result that is **independent of hyperedge size** in general

Hyper-flow diffusion: local clustering guarantee

Conductance of target cluster C

$$\Phi(C) = \frac{\sum_{e \in E} w_e(C)}{\min\{\mathbf{vol}(C), \mathbf{vol}(V \backslash C)\}}$$ where $\mathbf{vol}(C) := \sum_{v \in C} d(v)$

Seed set $S := \mathbf{supp}(\Delta)$.

Assumption 1 (sufficient overlap):
$$\mathbf{vol}(S \cap C) \geq \beta \mathbf{vol}(S)$$
$$\mathbf{vol}(S \cap C) \geq \alpha \mathbf{vol}(C)$$
$\alpha, \beta \geq \frac{1}{\log^t \mathbf{vol}(C)}$ for some t

Assumption 2: $0 \leq \sigma \leq \beta \Phi(C)/3$

The output cluster $\tilde{C}$ satisfies $\Phi(\tilde{C}) \leq \tilde{\mathcal{O}}(\sqrt{\Phi(C)})$

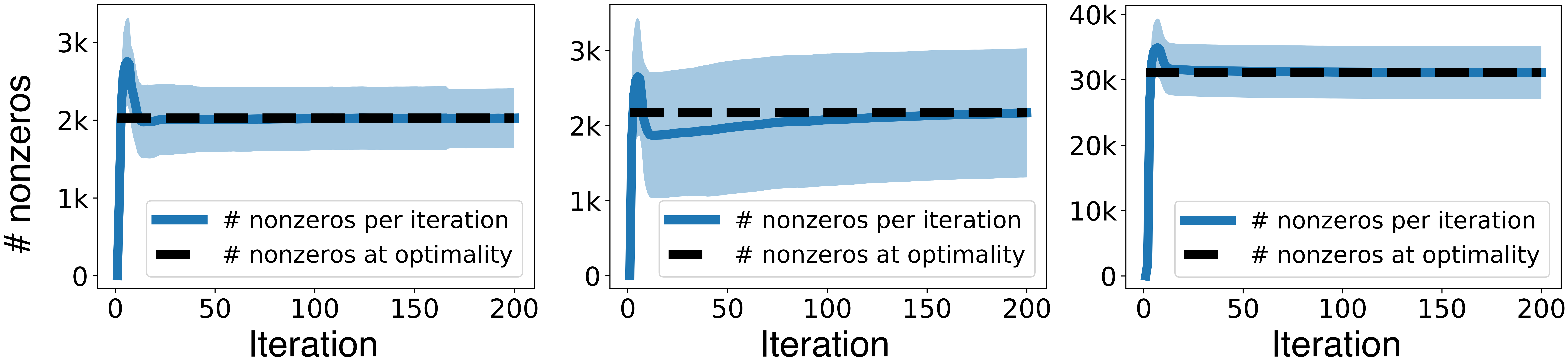
The first result that is **independent of hyperedge size** in general

An important part of the proof builds on a **generalized Rayleigh quotient lower bound** for hypergraphs

Hyper-flow diffusion: algorithm

We solve an equivalent primal reformulation via **alternating minimization**.

The algorithm only touches a small part of the hypergraph.

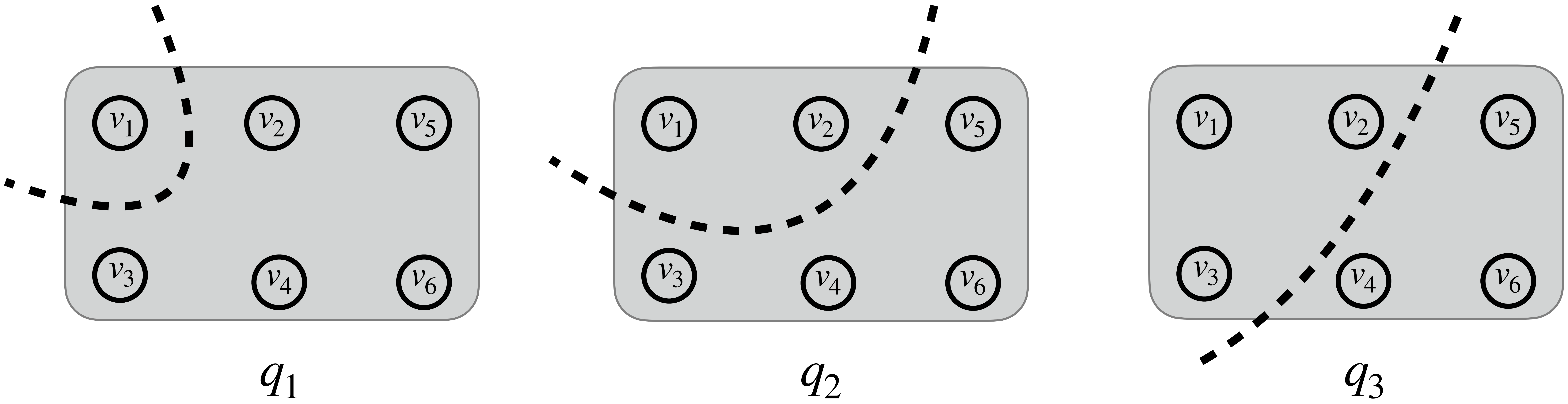


The figures show the number of nodes touched by the algorithm on 3 different clusters in the Amazon-reviews dataset, which consists of 2.2 million nodes.

Hyper-flow diffusion: empirical results

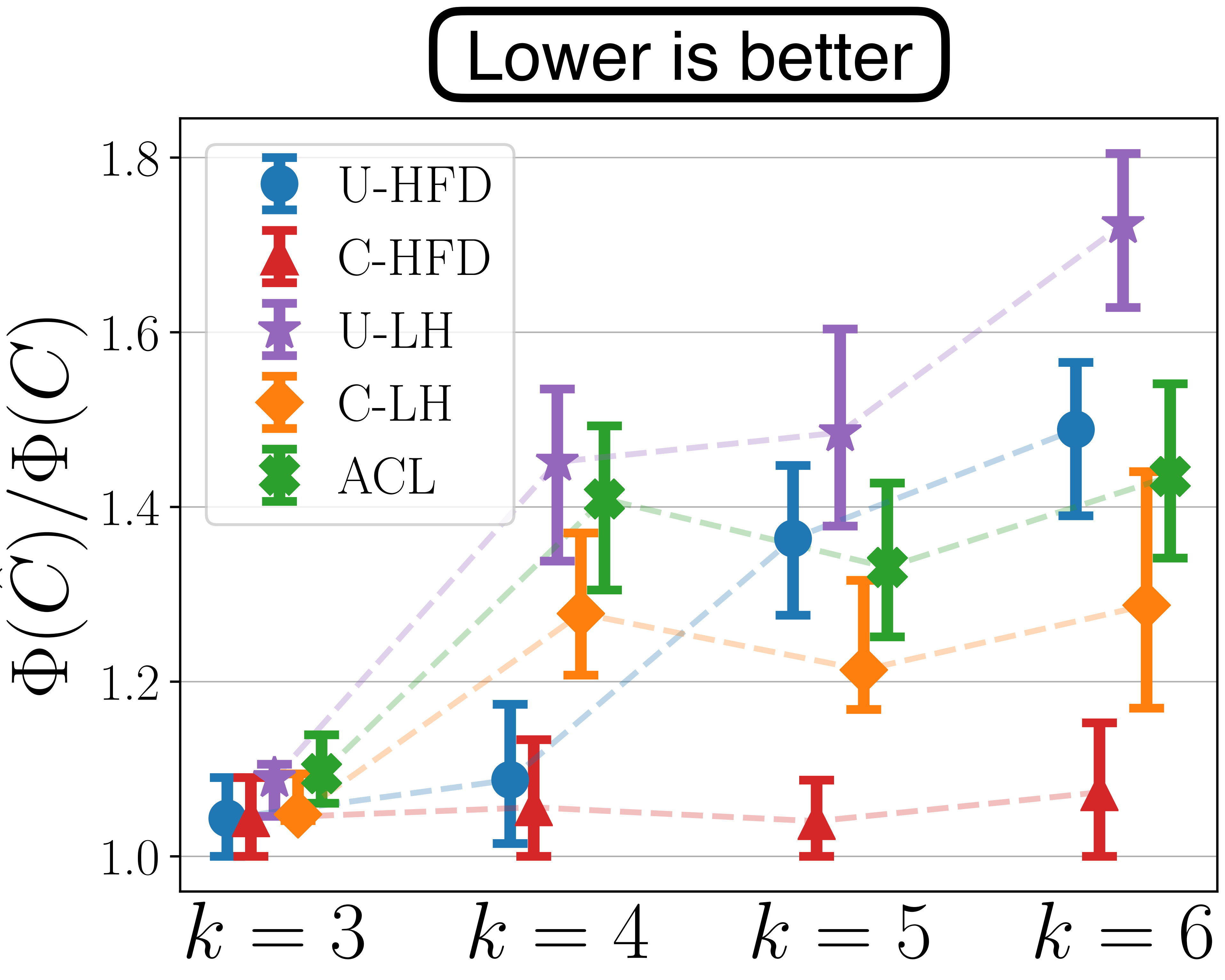
Cardinality-based k-uniform stochastic block model:

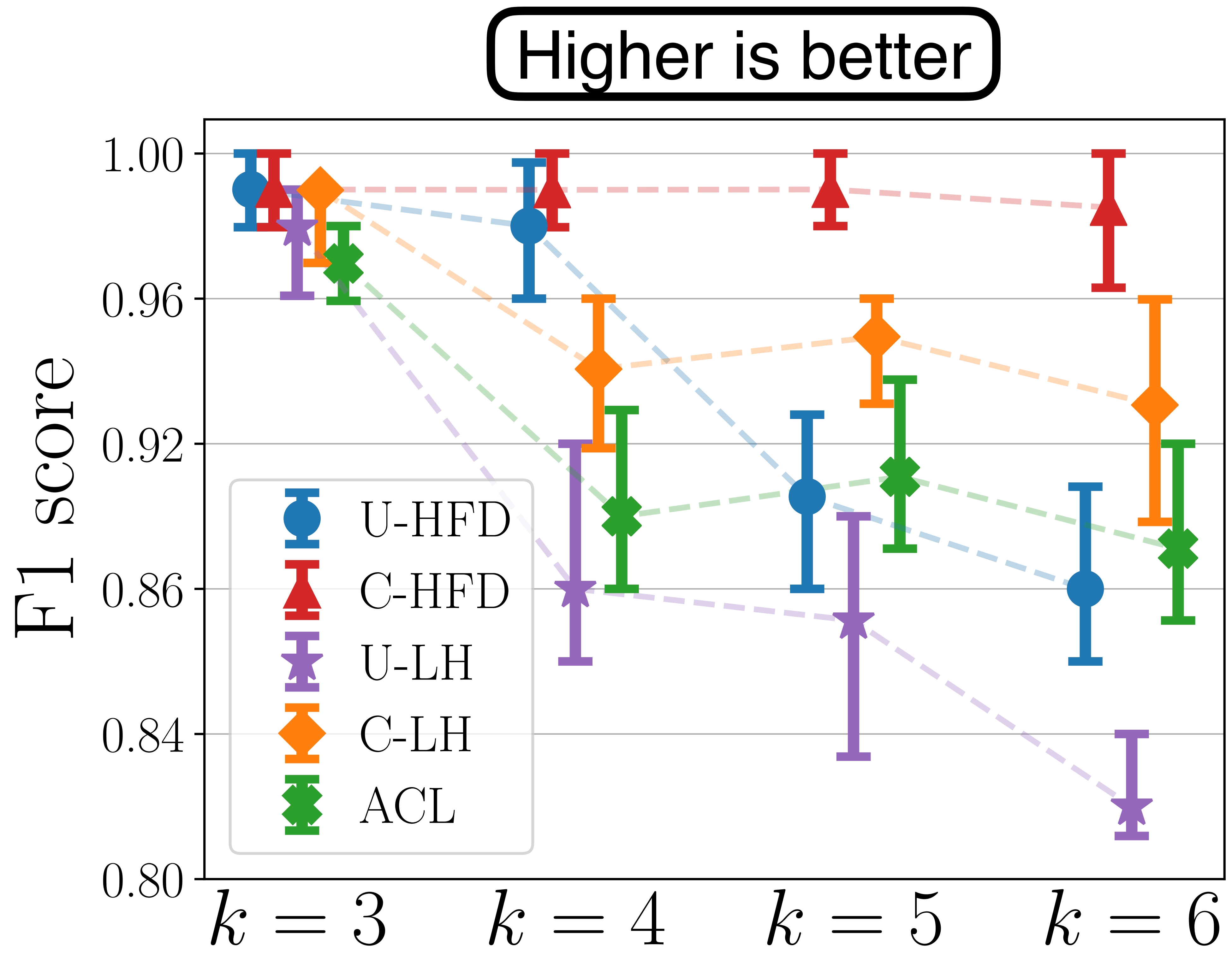
Boundary hyperedges appear with different probabilities according to the cardinality of hyperedge cut.



We consider $q_1 \gg q_2 \geq q_3$. Under this generative setting, one should naturally explore cardinality-based cut-cost for clustering.

Hyper-flow diffusion: empirical results





U-* means unit cut-cost; C-* means cardinality-based cut-cost.

For each method, C-* is better than U-*.

There is a significant performance drop for C-LH at $k = 4$.

Hyper-flow diffusion: empirical results

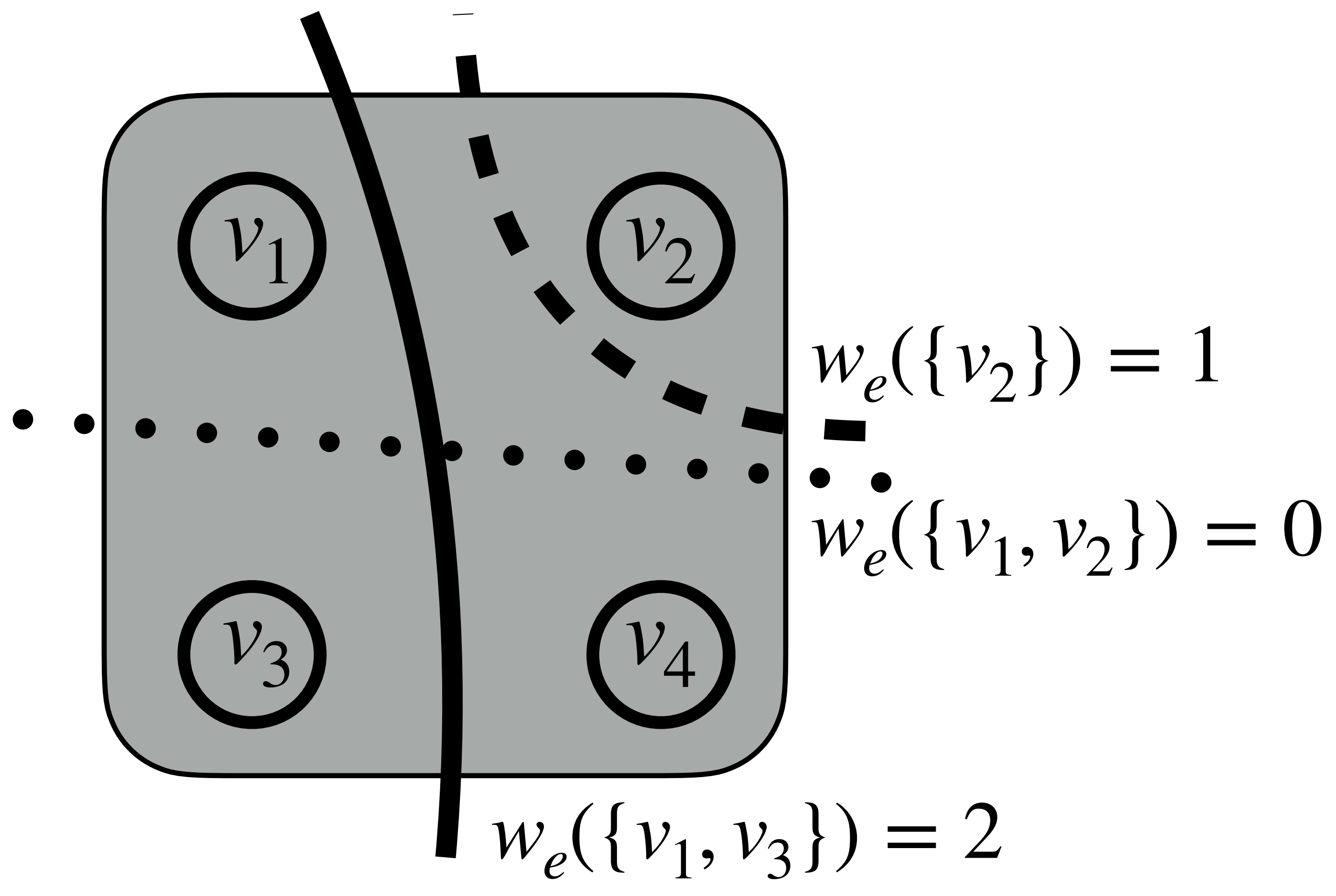
F1 scores for local clustering on a real hypergraph constructed from travel metasearch data.

Method	South Korea	Iceland	Puerto Rico	Crimea	Vietnam	Hong Kong	Malta	Guatemala	Ukraine	Estonia
U-HFD	0.75	**0.99**	0.89	0.85	0.28	0.82	**0.98**	0.94	0.60	**0.94**
C-HFD	**0.76**	**0.99**	**0.95**	**0.94**	**0.32**	0.80	**0.98**	**0.97**	**0.68**	**0.94**
U-LH-2.0	0.70	0.86	0.79	0.70	0.24	0.92	0.88	0.82	0.50	0.90
C-LH-2.0	0.73	0.90	0.84	0.78	0.27	**0.94**	0.96	0.88	0.51	0.83
U-LH-1.4	0.69	0.84	0.80	0.75	0.28	0.87	0.92	0.83	0.47	0.90
C-LH-1.4	0.71	0.88	0.84	0.78	0.27	0.88	0.93	0.85	0.50	0.85
ACL	0.65	0.84	0.75	0.68	0.23	0.90	0.83	0.69	0.50	0.88

Hyper-flow diffusion: empirical results

Node-ranking and and local clustering results on a Florida Bay food network.

	Top-2 node-ranking results		Clustering F1		
Method	Query: Raptors	Query: Gray Snapper	Prod.	Low	High
U-HFD	Epiphytic Gastropods, Detriti. Gastropods	Meiofauna, Epiphytic Gastropods	**0.69**	0.47	0.64
C-HFD	Predatory Shrimp, Herbivorous Shrimp	Herb. Amphipods, Pink Shrimp	0.67	0.53	0.43
S-HFD	Gruiformes, Small Shorebirds	Snook, Mojarra	**0.69**	**0.65**	**0.83**



S-HFD uses specialized submodular cut-cost shown on the left.

The example shows that general submodular cut-cost can be necessary.

Thank you!

Hyper-flow diffusion: more empirical results

Conductance and F1 results for local clustering on real hypergraphs.
Unit cut-cost is used in these experiments.

Metric	Alg.	Amazon-reviews									Microsoft-academic				Florida-Bay		
		1	2	3	12	15	17	18	24	25	Data	ML	TCS	CV	Prod.	Low	High
Cond	HFD	**0.17**	**0.11**	**0.12**	**0.16**	**0.36**	**0.25**	**0.17**	**0.14**	**0.28**	**0.03**	**0.06**	**0.06**	**0.03**	**0.49**	**0.36**	**0.35**
	LH-2.0	0.42	0.50	0.25	0.44	0.74	0.44	0.57	0.58	0.61	0.07	0.09	0.10	0.07	0.51	0.39	0.39
	LH-1.4	0.33	0.44	0.25	0.36	0.81	0.40	0.51	0.54	0.59	0.07	0.08	0.09	0.07	**0.49**	0.39	0.41
	ACL	0.42	0.50	0.25	0.54	0.77	0.52	0.63	0.68	0.65	0.08	0.11	0.11	0.09	0.52	0.39	0.40
F1	HFD	**0.45**	**0.09**	**0.65**	**0.92**	0.04	**0.10**	**0.80**	**0.81**	**0.09**	**0.78**	**0.54**	**0.86**	**0.73**	**0.69**	**0.47**	**0.64**
	LH-2.0	0.23	0.07	0.23	0.29	**0.05**	0.06	0.21	0.28	0.05	0.67	0.46	0.71	0.61	**0.69**	0.45	0.57
	LH-1.4	0.23	**0.09**	0.35	0.40	0.00	0.07	0.31	0.35	0.06	0.65	0.46	0.59	0.59	**0.69**	0.45	0.58
	ACL	0.23	0.07	0.22	0.25	0.04	0.05	0.17	0.20	0.04	0.64	0.43	0.70	0.57	**0.69**	0.44	0.57