Hyper-Flow Diffusion

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

[1]University of Waterloo [2]Purdue University

Networks 2021

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

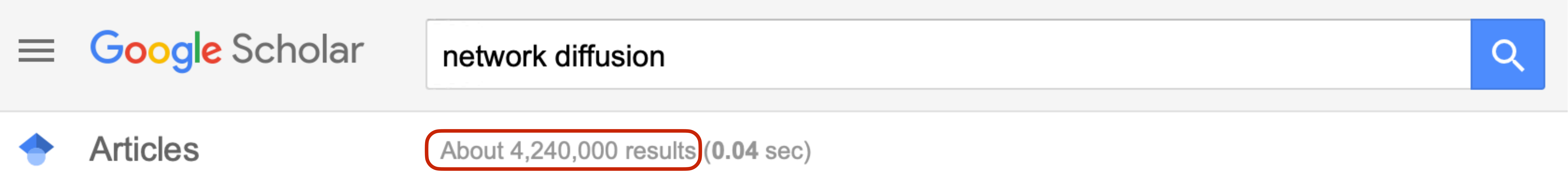
Collaboration

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

Ecology

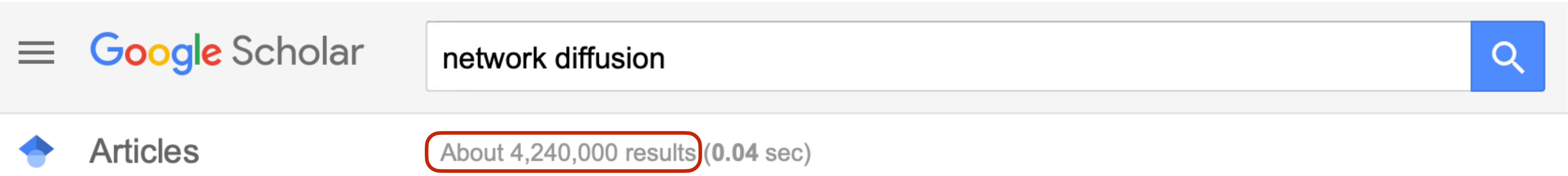
Nodes are species
Multiple species interact according to their roles in the food chain

Diffusion algorithms are everywhere (for graphs)



4.2 million results

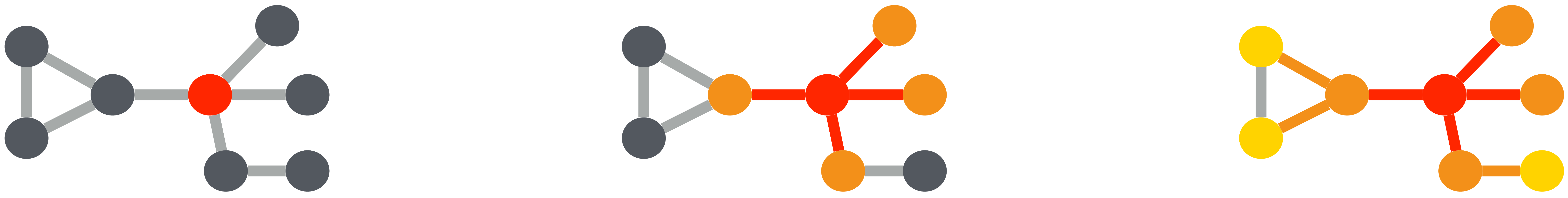
Diffusion algorithms are everywhere (for graphs)



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.

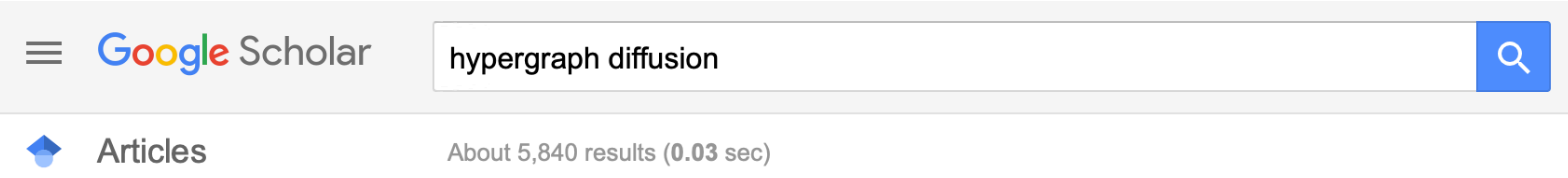
Applications include *recommendation systems*, *node ranking*, *community detection*, *social and biological network analysis*, etc.



1 *2* *3*

Diffusion algorithms are everywhere (for graphs)

4.2 million results



However … 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 to large datasets.

Our motivation

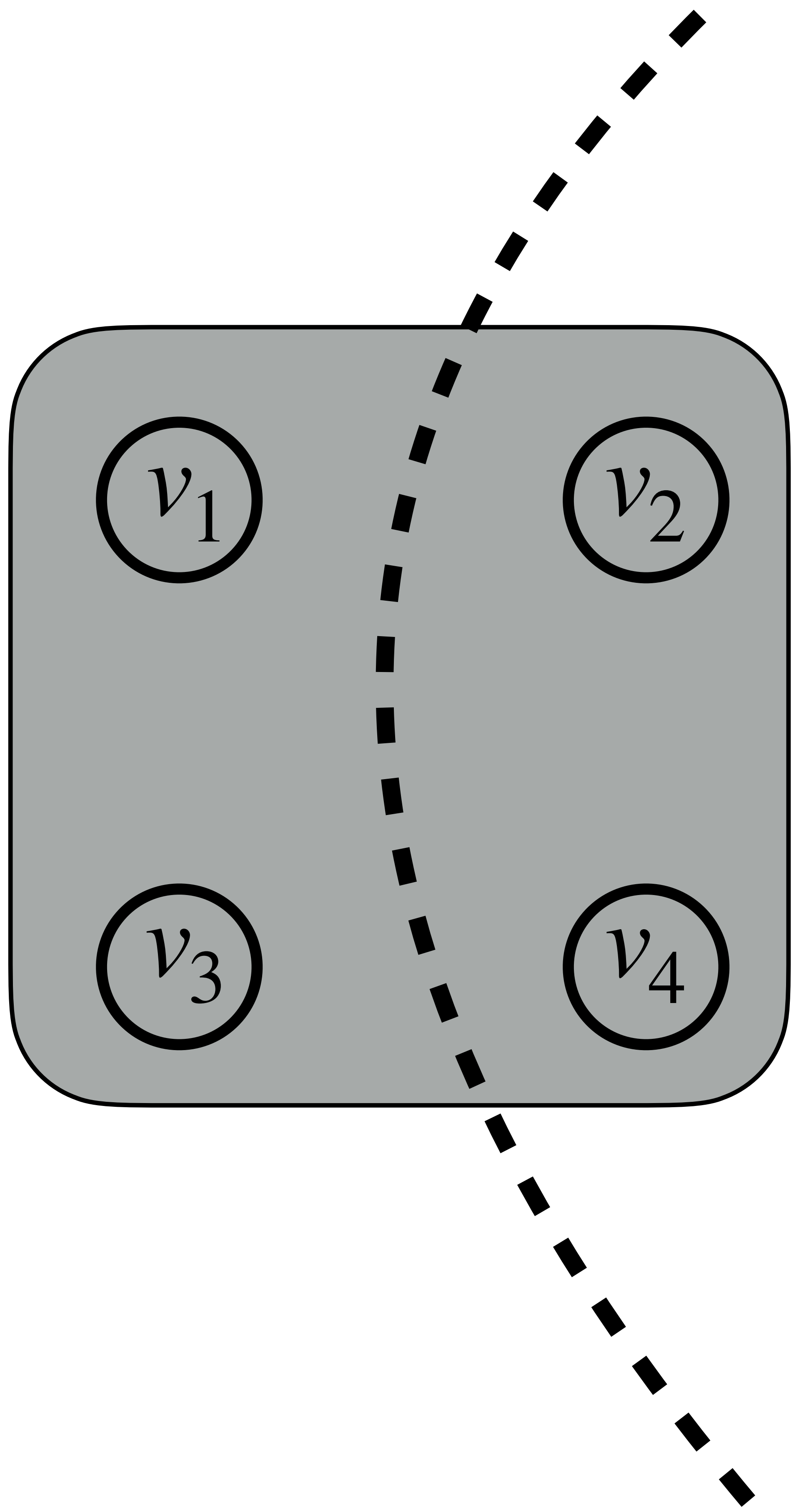
We propose the first local diffusion method that

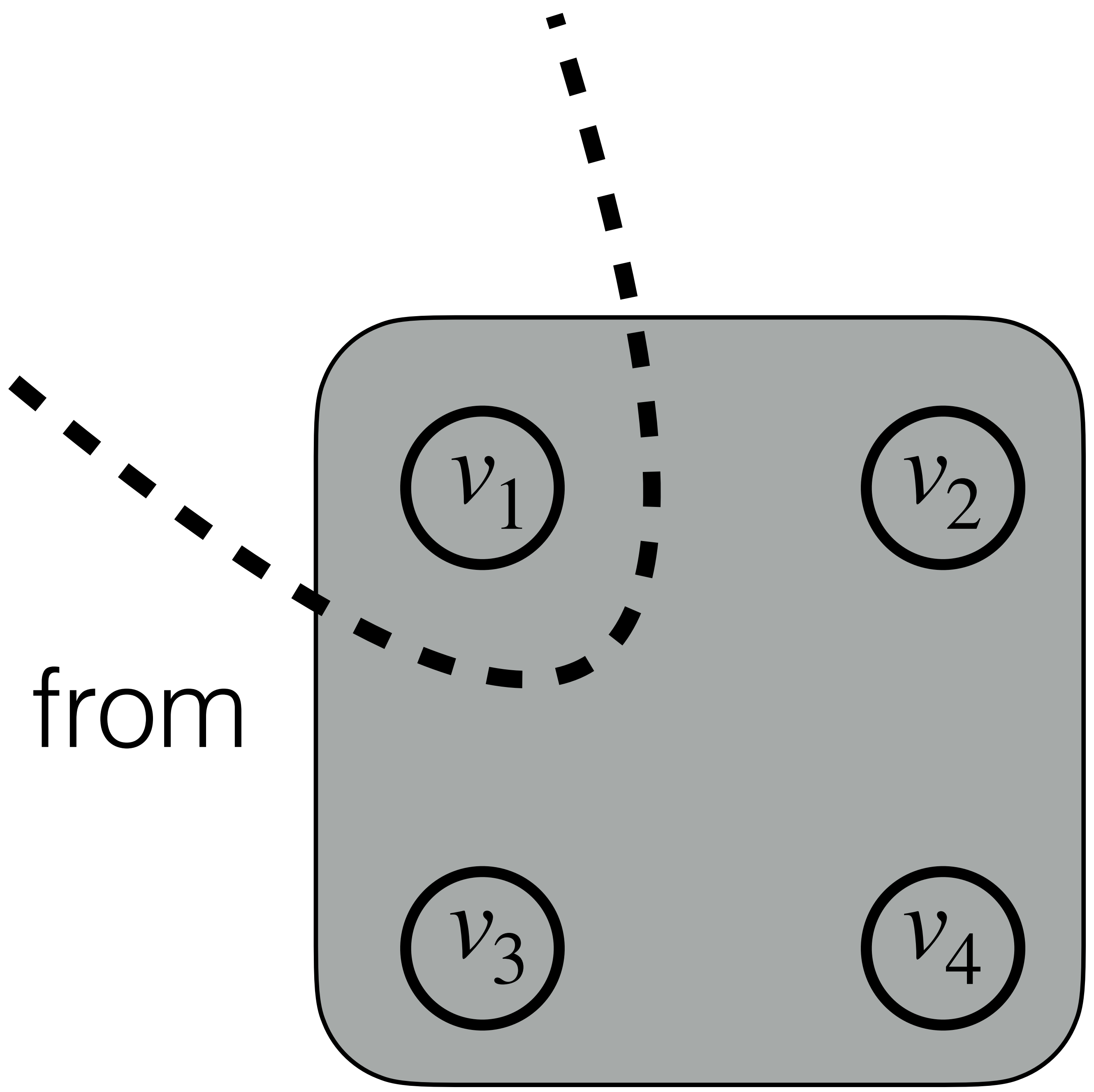
- Achieves stronger theoretical guarantees for the local hypergraph clustering problem;
- Applies to a substantially richer class of higher-order relations with only a submodularity assumption;
- Permits computational efficient algorithms.

However … 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** to large datasets.

Higher-order relations: hyperedge cut perspective

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

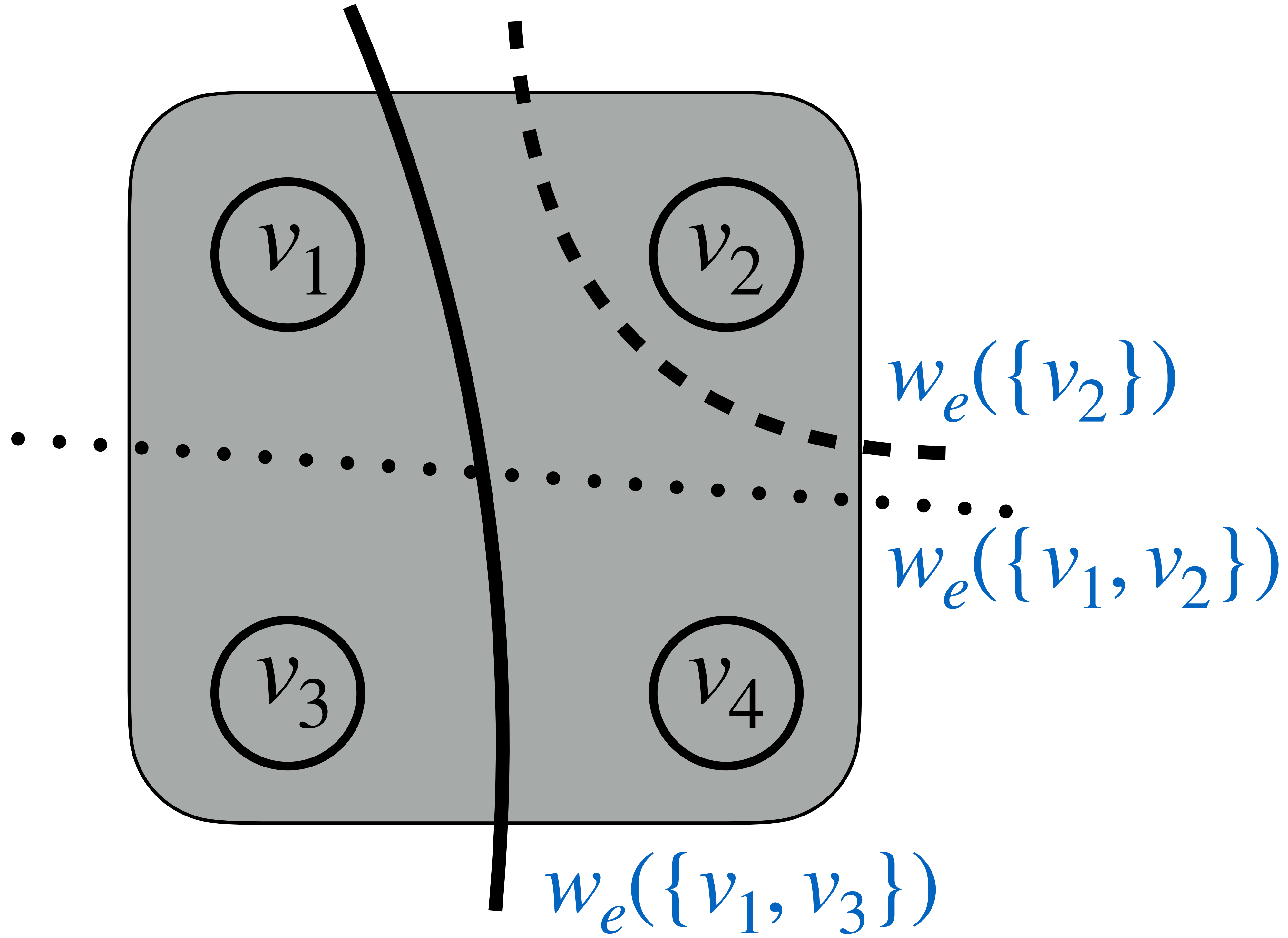
How do we treat 

differently from 

?

Higher-order relations: hyperedge cut perspective

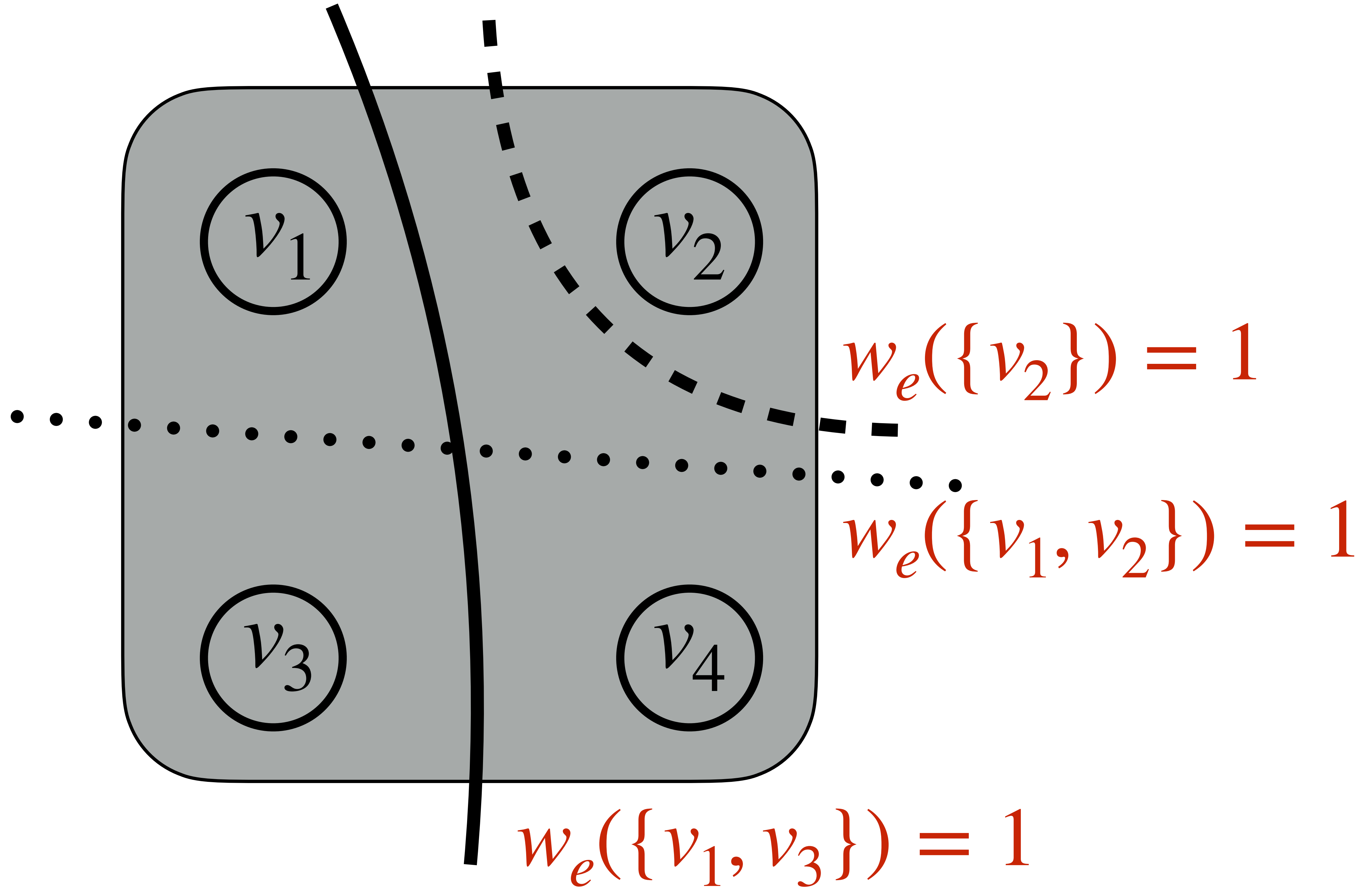
Distinct ways to cut a 4-node hyperedge may have different costs.



$w_e(S)$ 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.

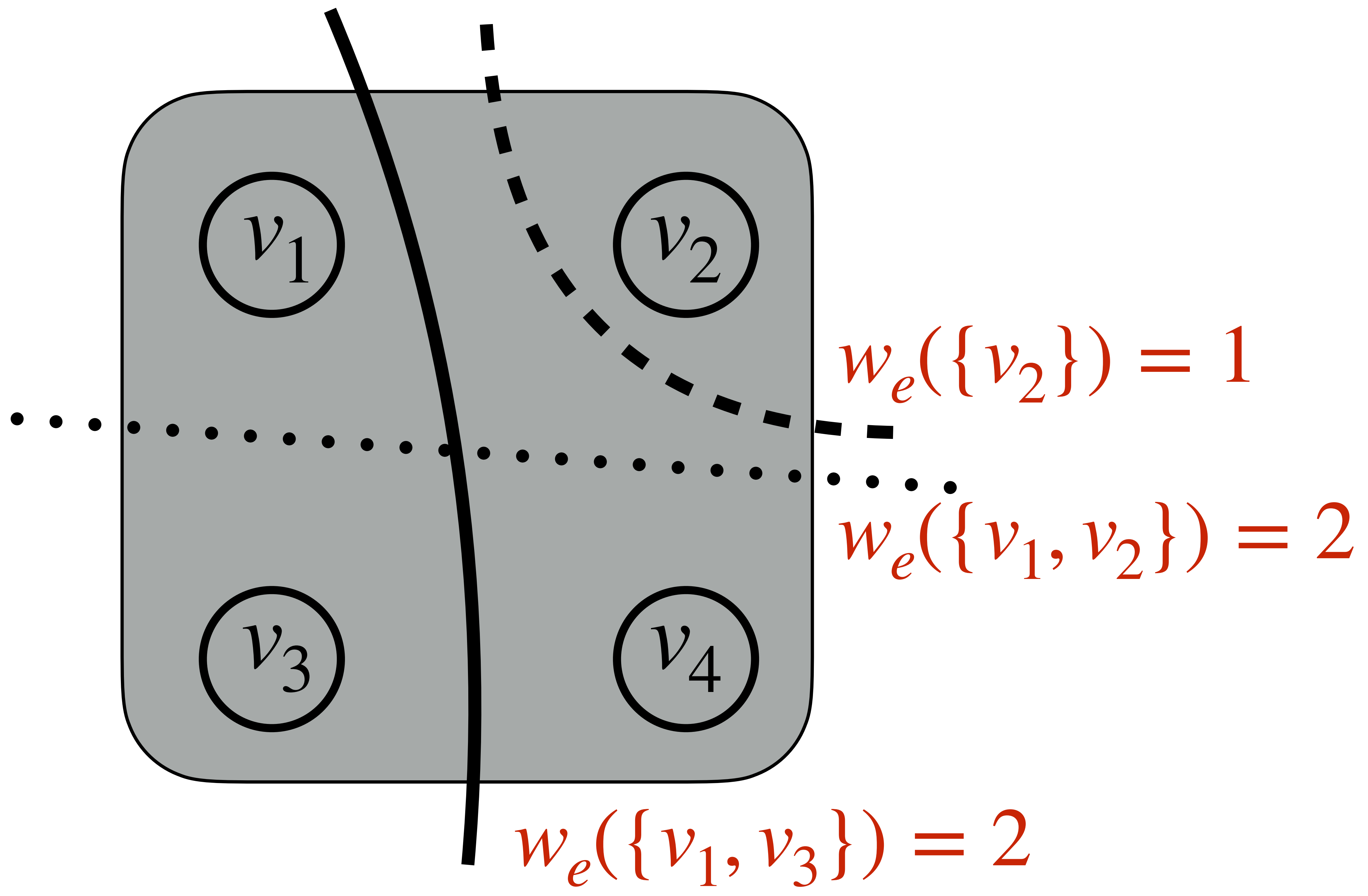


$w_e(S)$ specifies the cost of splitting e into S and $e \backslash S$.

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.



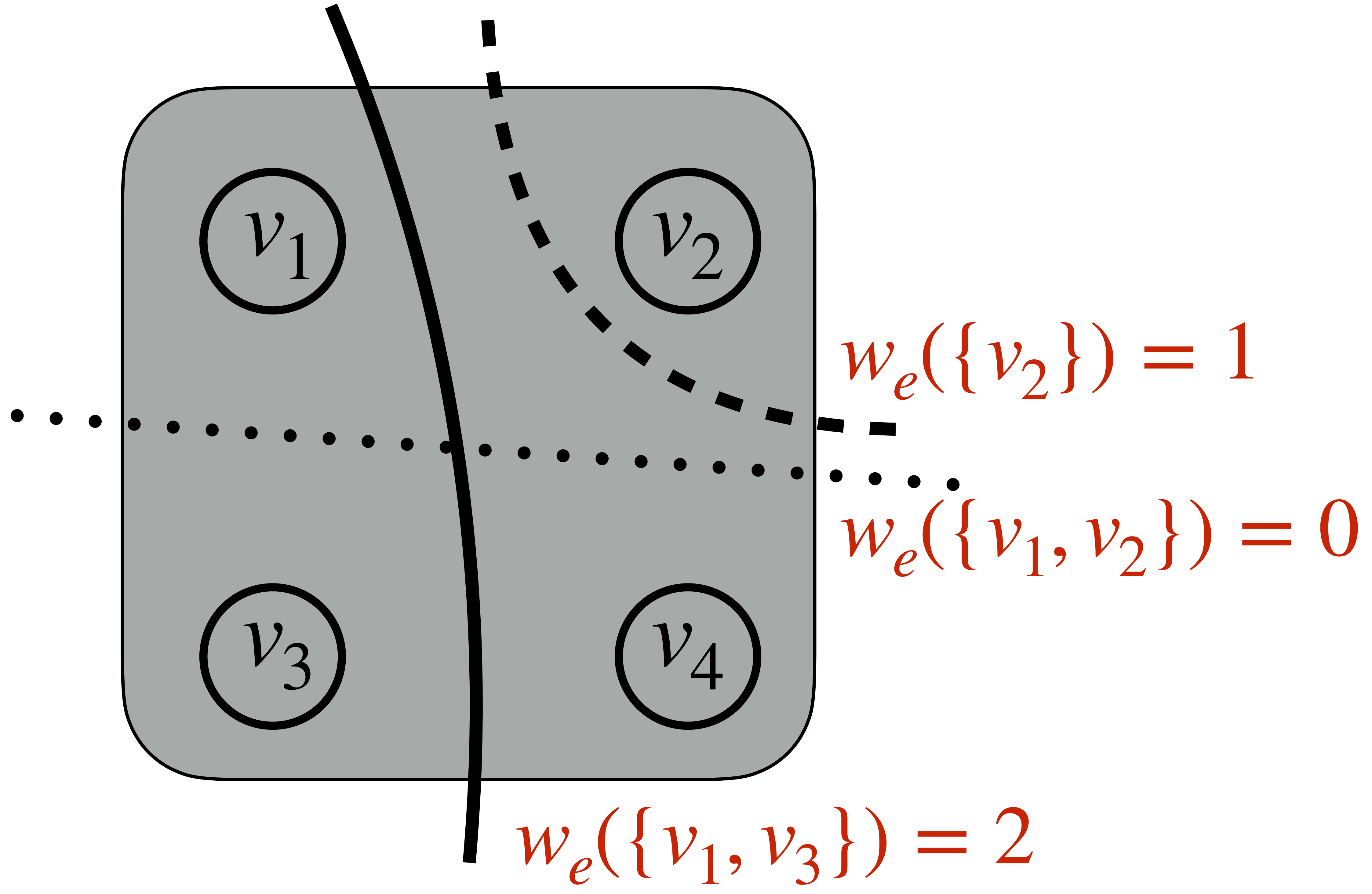
$w_e(S)$ specifies the cost of splitting e into S and $e \backslash S$.

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.



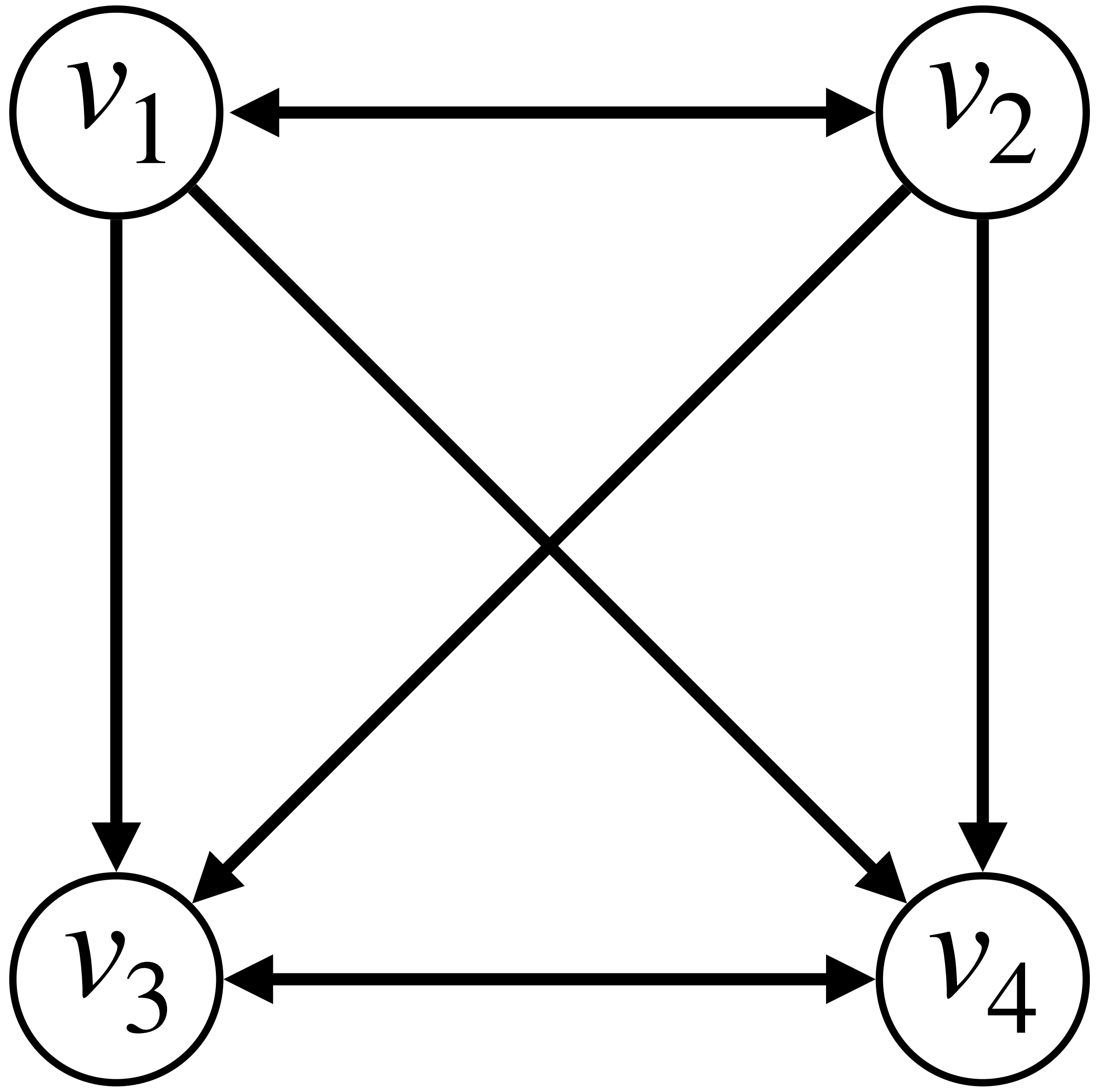
$w_e(S)$ specifies the cost of splitting e into S and $e \backslash S$.

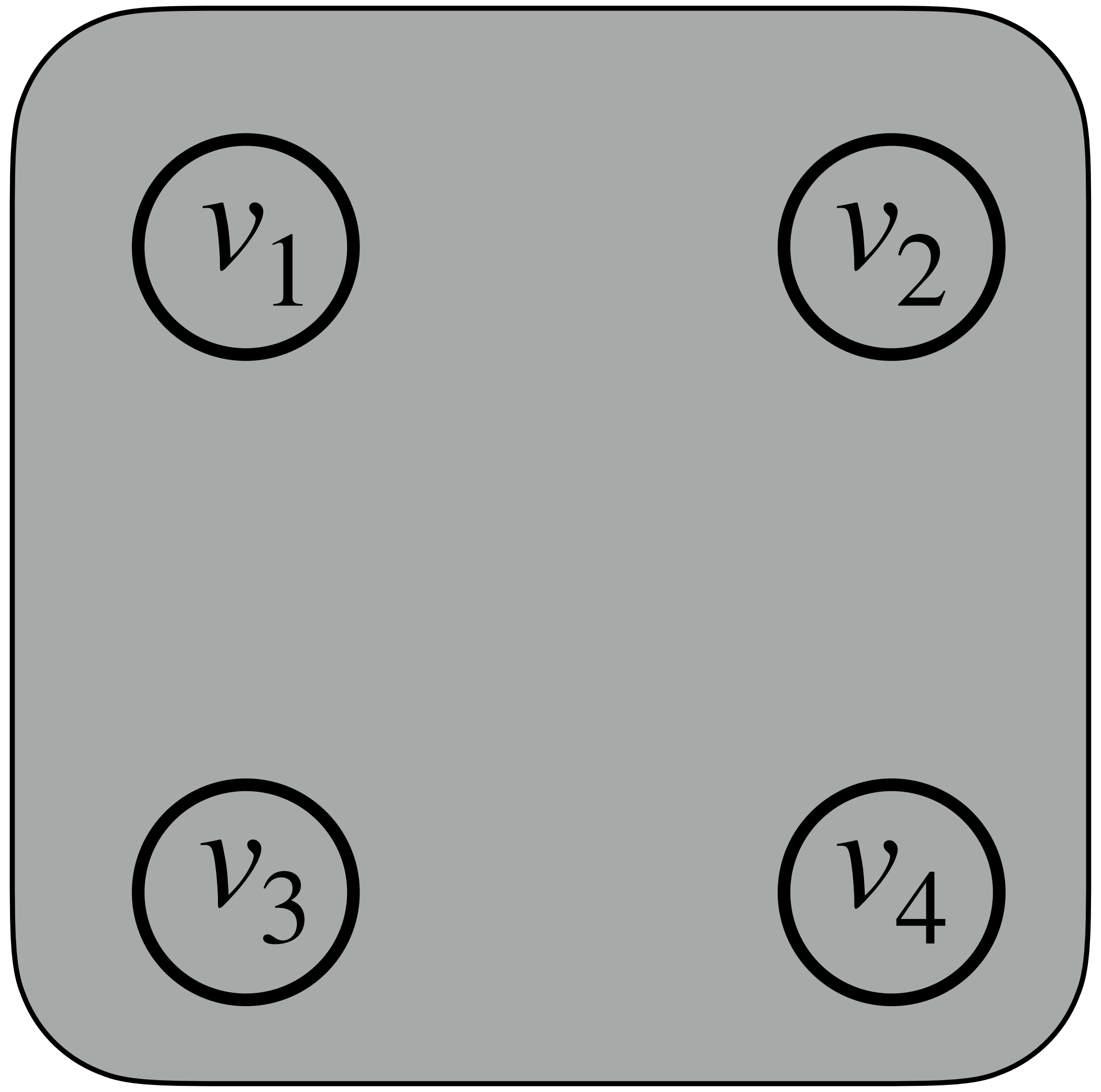
Unit: the cost of cutting a hyperedge is always 1, i.e., $w_e(S) = 1$.

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., $w_e : 2^e \rightarrow \mathbb{R}$ is a submodular set function.

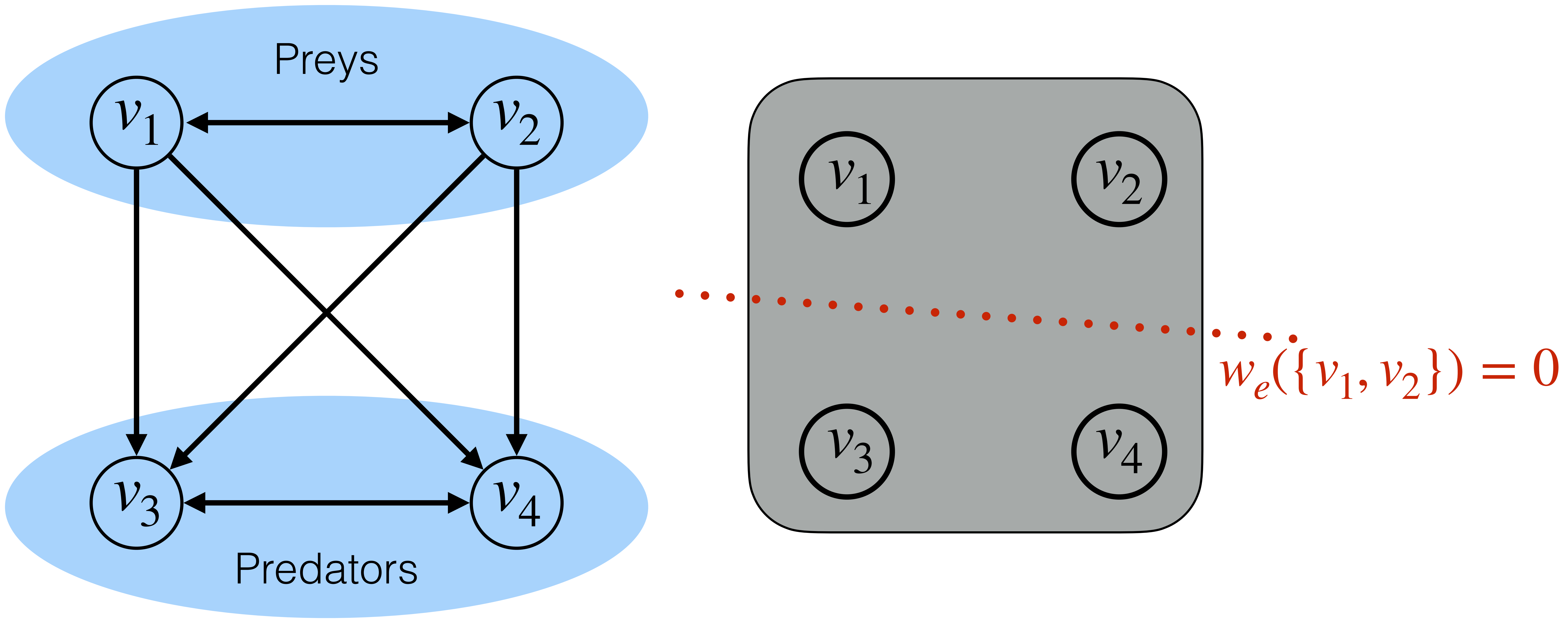
Higher-order relations: hyperedge cut perspective





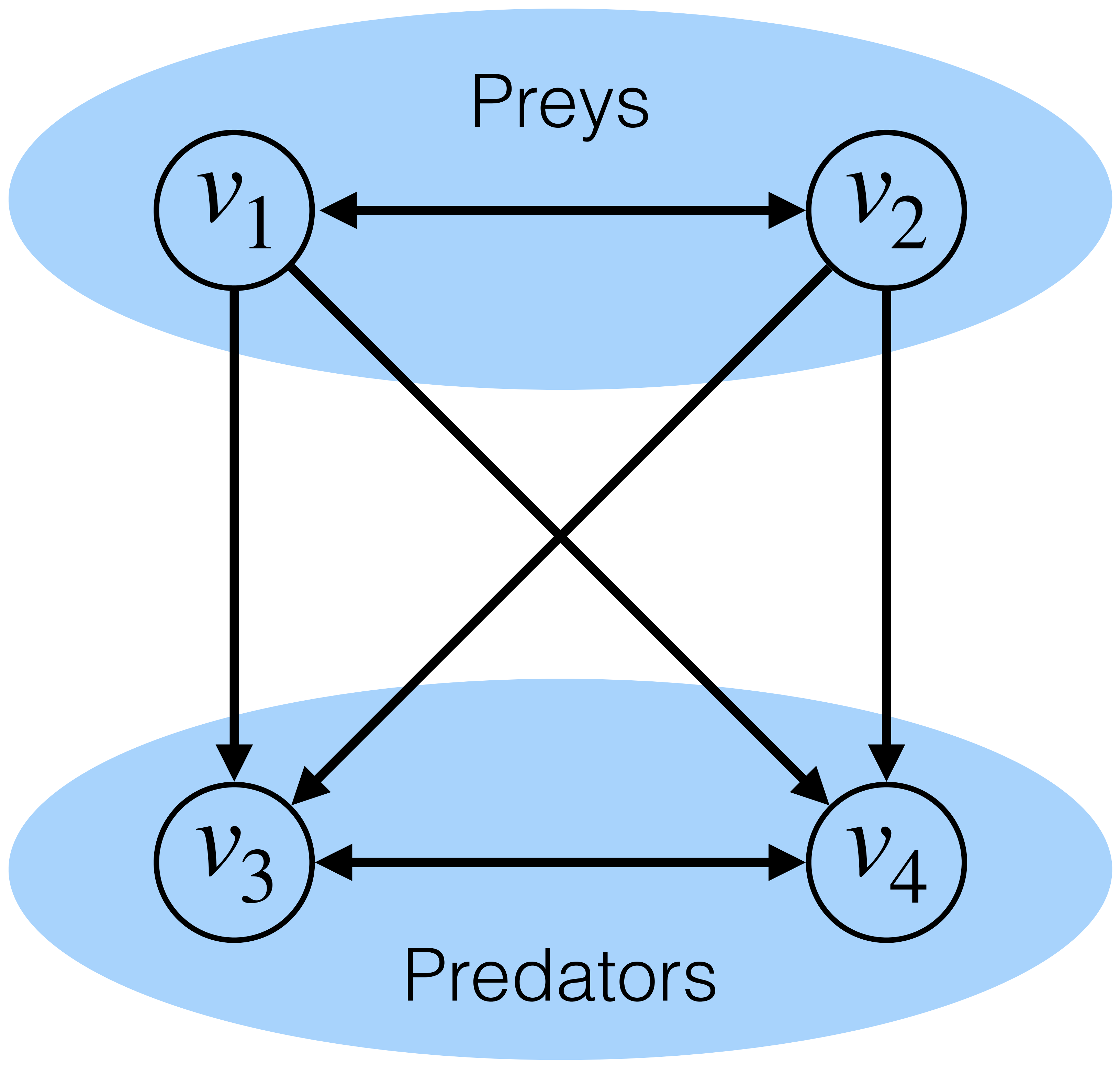
A food network can be mapped into a hypergraph by taking each network pattern on the left as a hyperedge on the right. This network pattern captures carbon flow from two preys (v_1, v_2) to two predators (v_3, v_4).

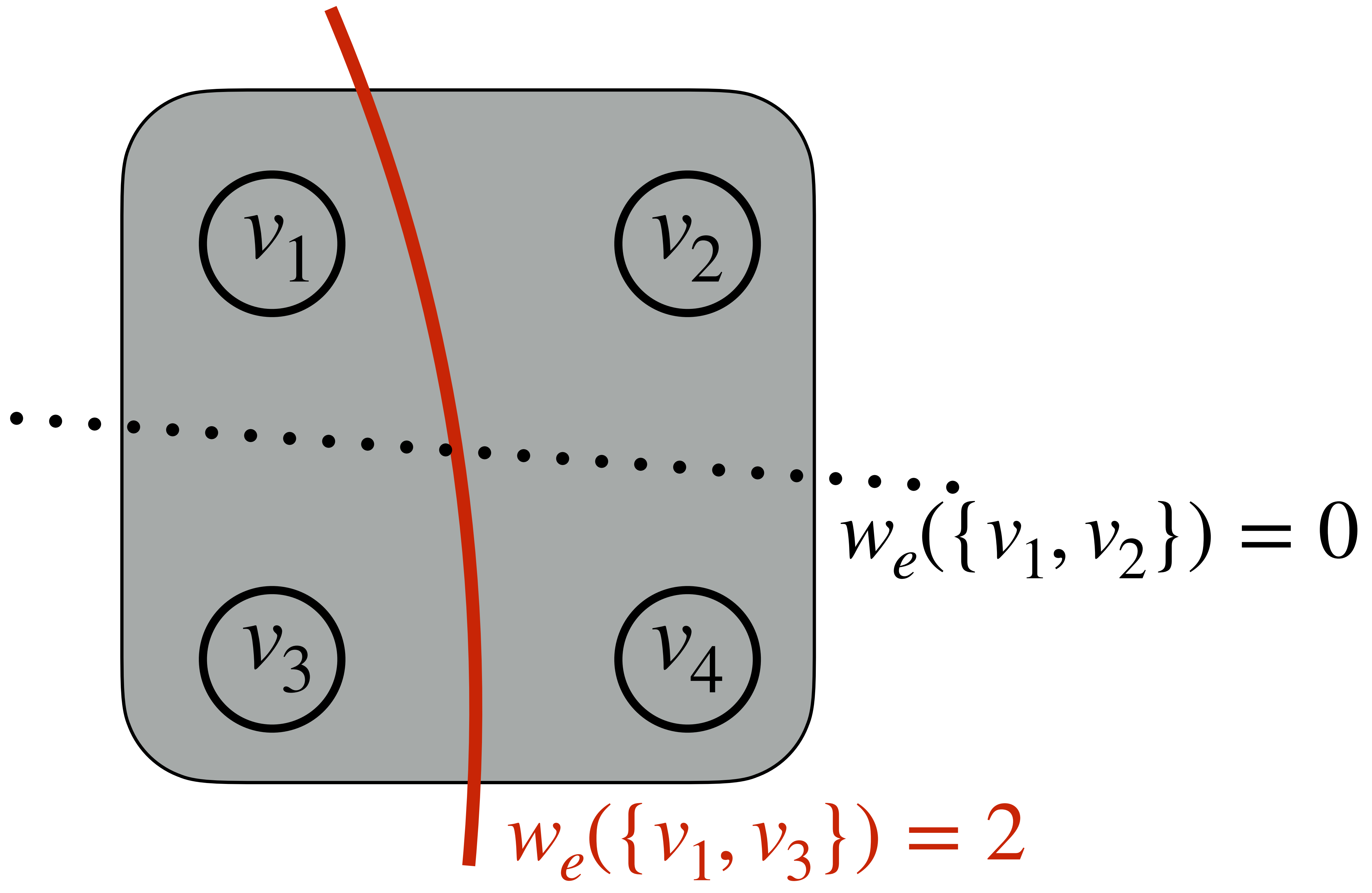
Higher-order relations: hyperedge cut perspective



The cut-cost $w_e(\{v_1, v_2\}) = w_e(\{v_3, v_4\}) = 0$ encourages separation of predators and preys.

Higher-order relations: hyperedge cut perspective

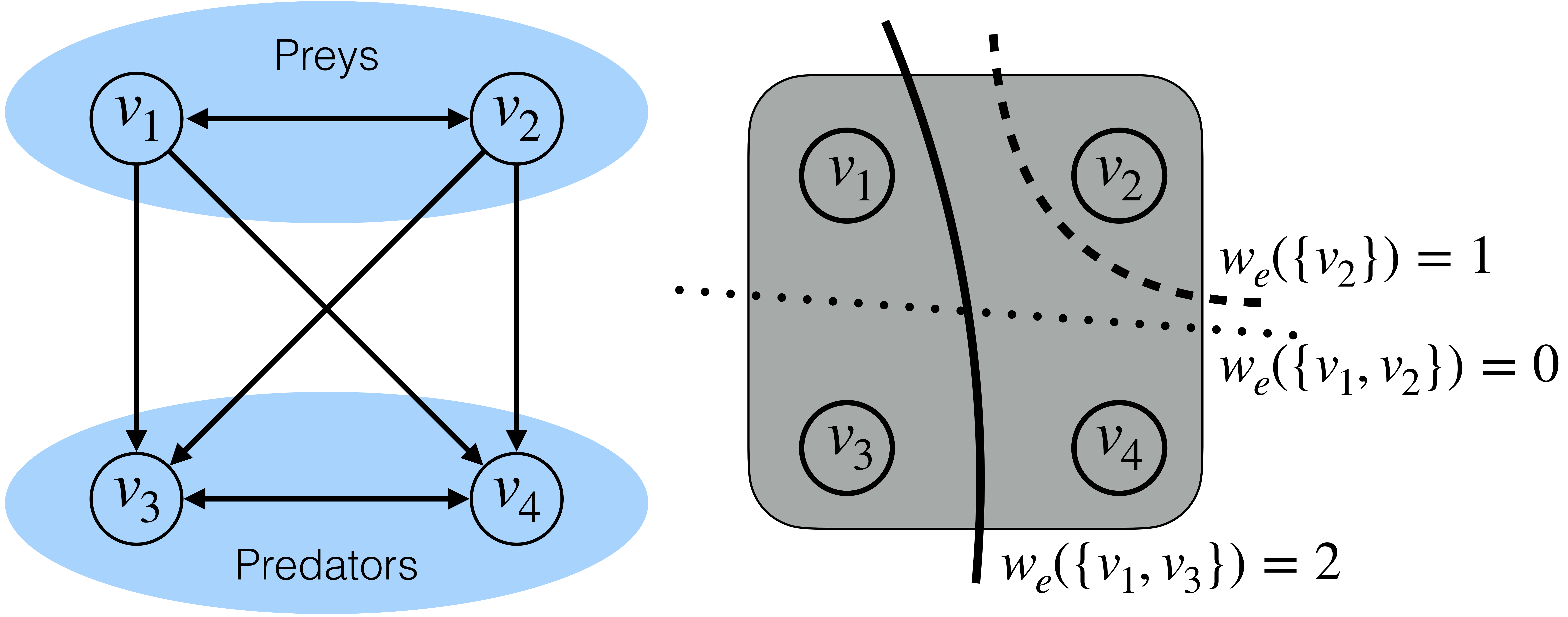




The cut-cost $w_e(\{v_1, v_2\}) = w_e(\{v_3, v_4\}) = 0$ encourages separation of predators and preys.

The cut-cost $w_e(\{v_1, v_3\}) = w_e(\{v_2, v_4\}) = 2$ discourages grouping of predators and preys.

Higher-order relations: hyperedge cut perspective

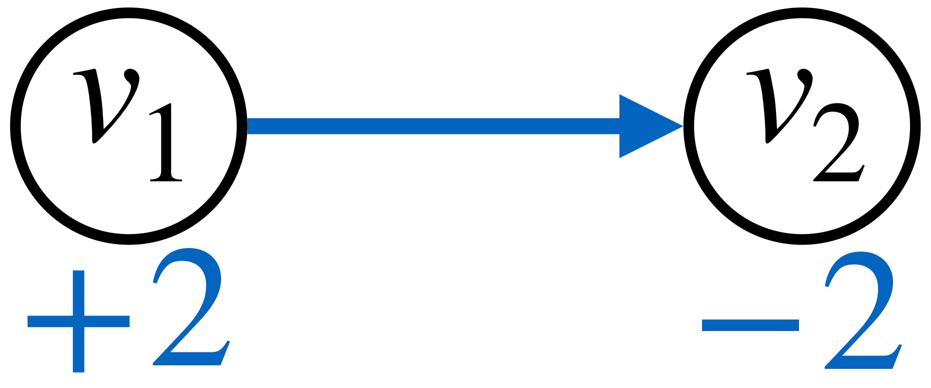


The cut-cost $w_e(\{v_1, v_2\}) = w_e(\{v_3, v_4\}) = 0$ encourages separation of predators and preys.

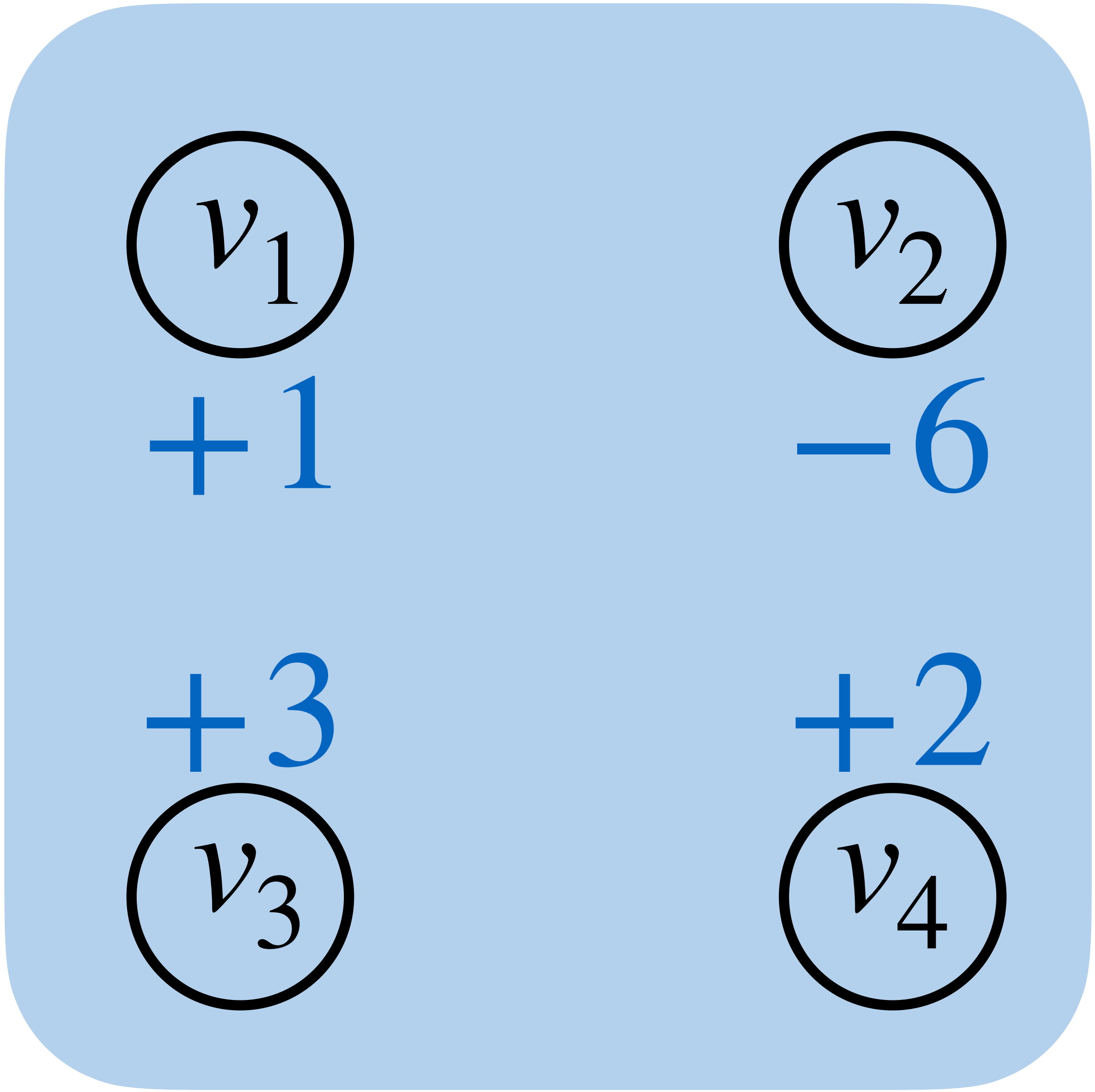
The cut-cost $w_e(\{v_1, v_3\}) = w_e(\{v_2, v_4\}) = 2$ discourages grouping of predators and preys.

The cut-cost $w_e(\{v_1\}) = w_e(\{v_2\}) = w_e(\{v_3\}) = w_e(\{v_4\}) = 1$ assigns less penalty for separating a single node. It also makes $w_e : 2^e \to \mathbb{R}_+$ a submodular function.

Higher-order relations: hyperedge flow perspective



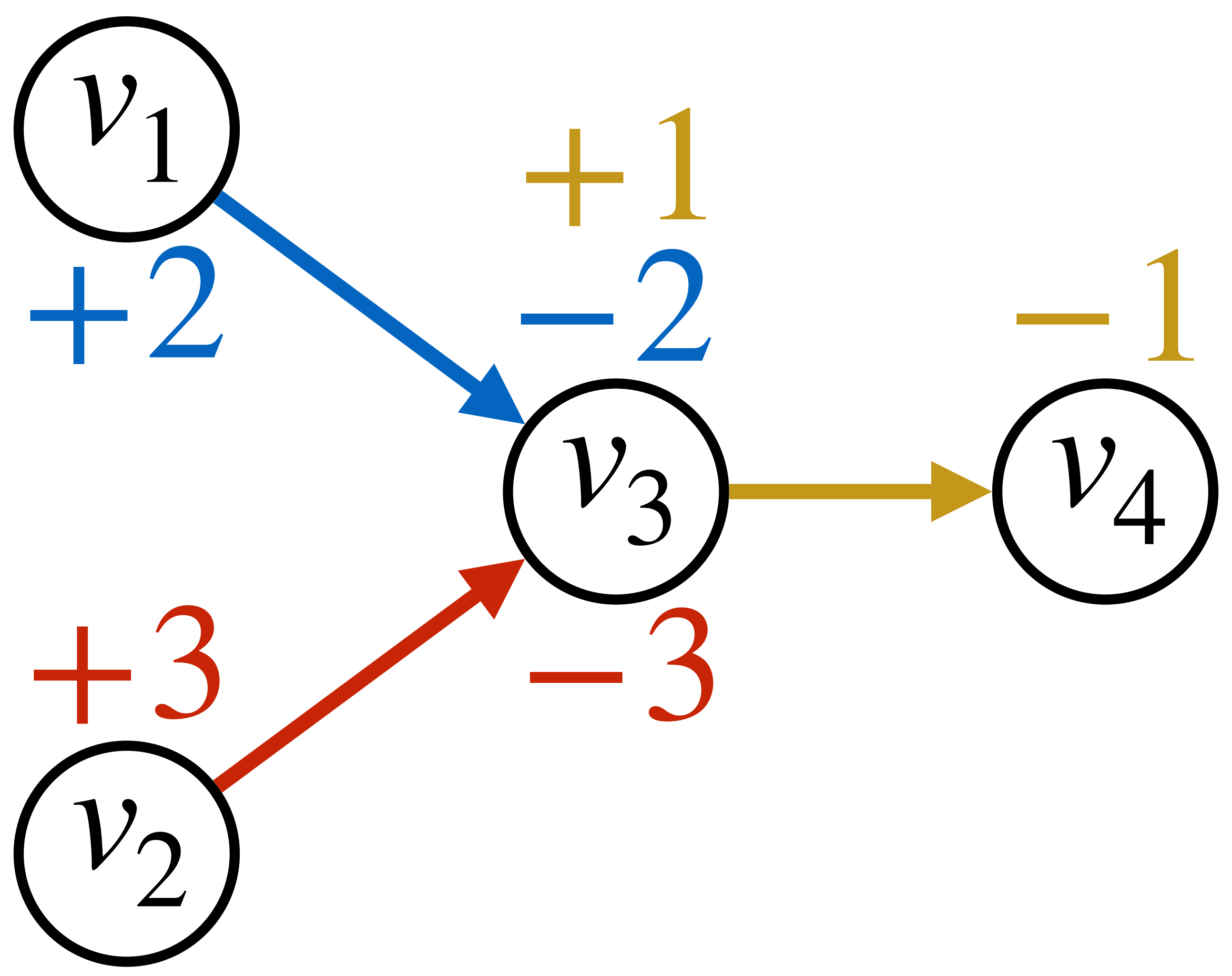
Graph edge



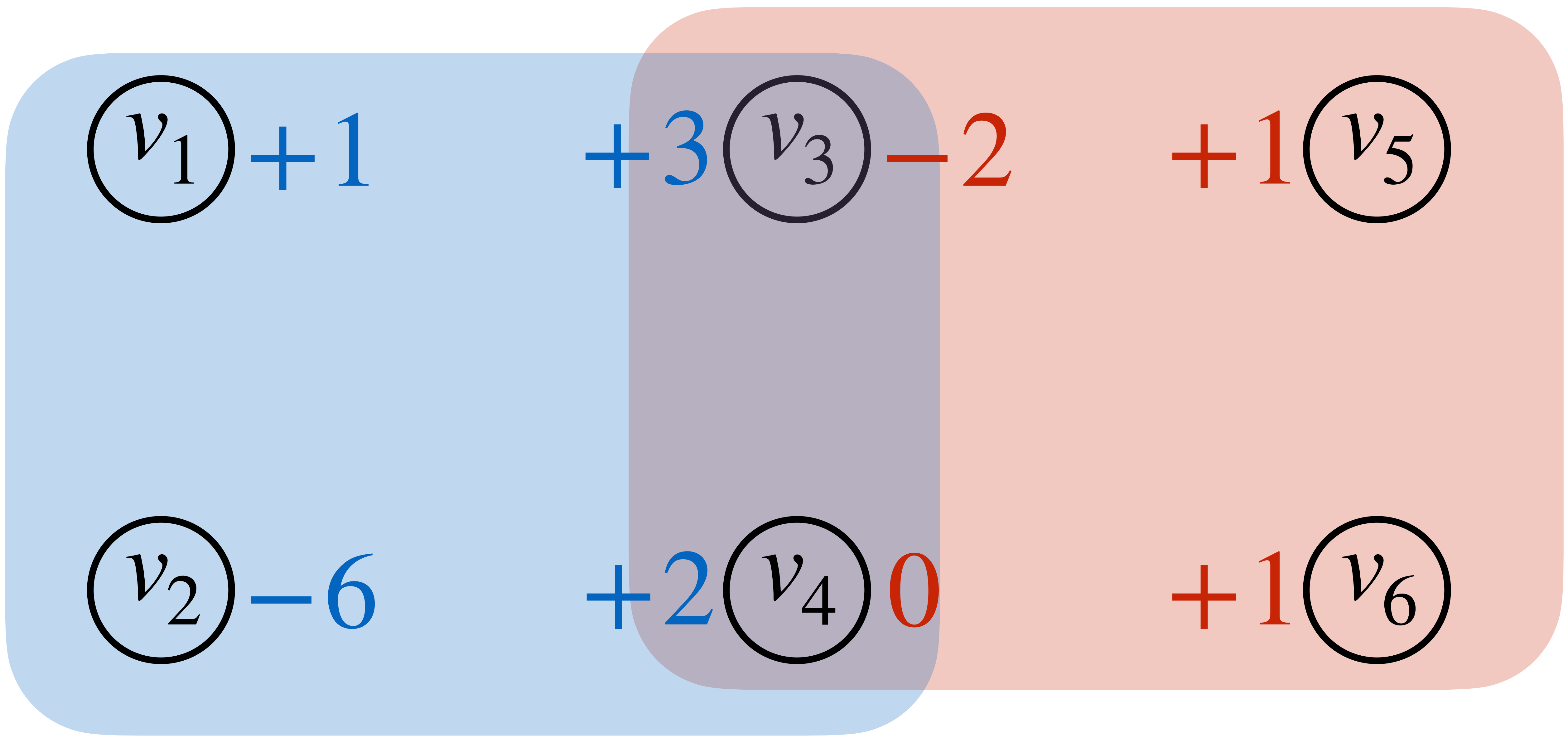
Hyperedge

To specify flows (i.e., movement of mass) over an edge or hyperedge, we associate each node a number which indicates the direction (sign) and magnitude of flow.

Higher-order relations: hyperedge flow perspective



Flows on graph



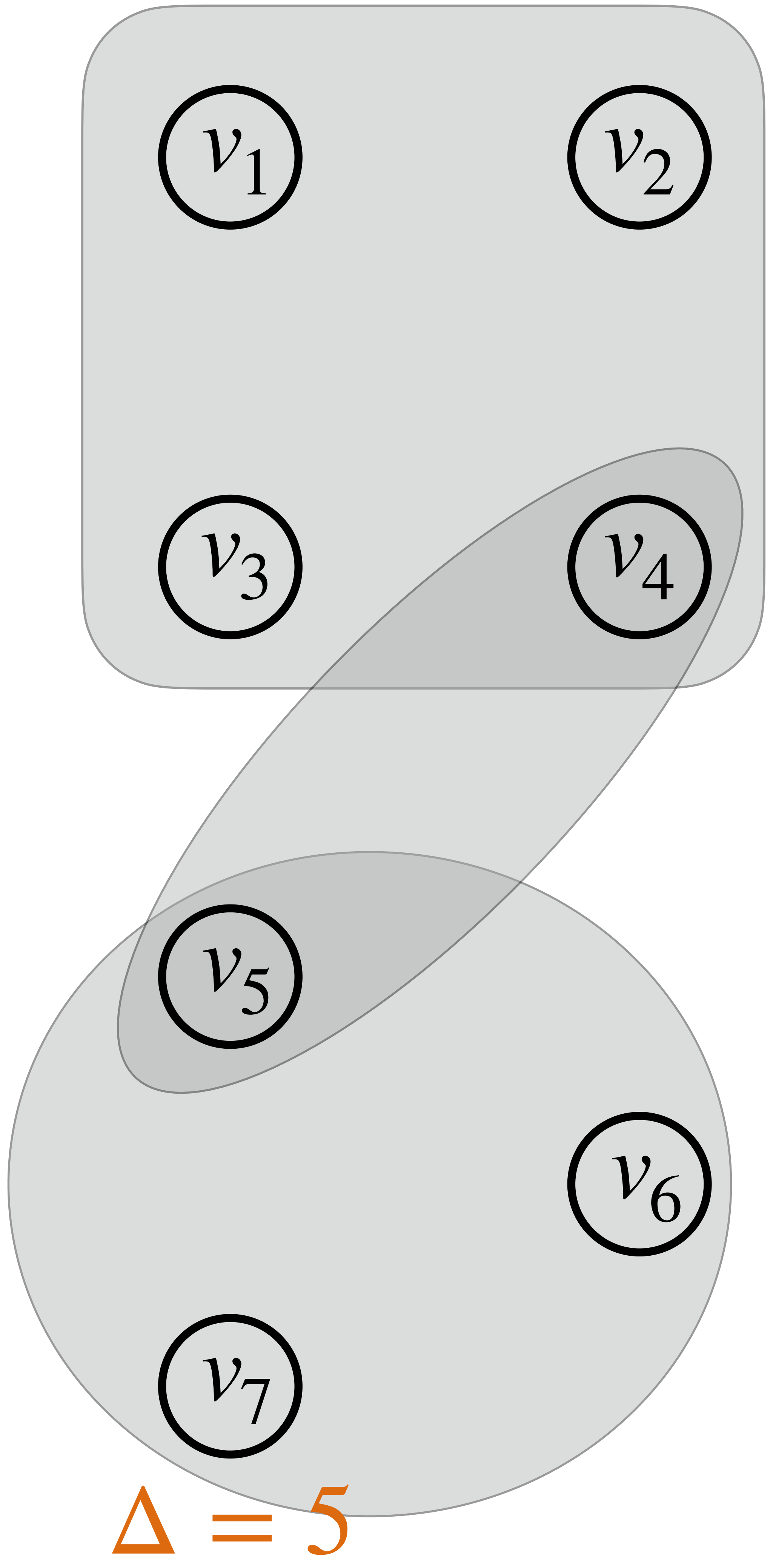
Flows on hypergraph

A natural generalization of network flows.

Flow conservation: numbers within the same hyperedge sum to 0.
Additional constraints required for hyperedges so that the numbers reflect higher-order relations.

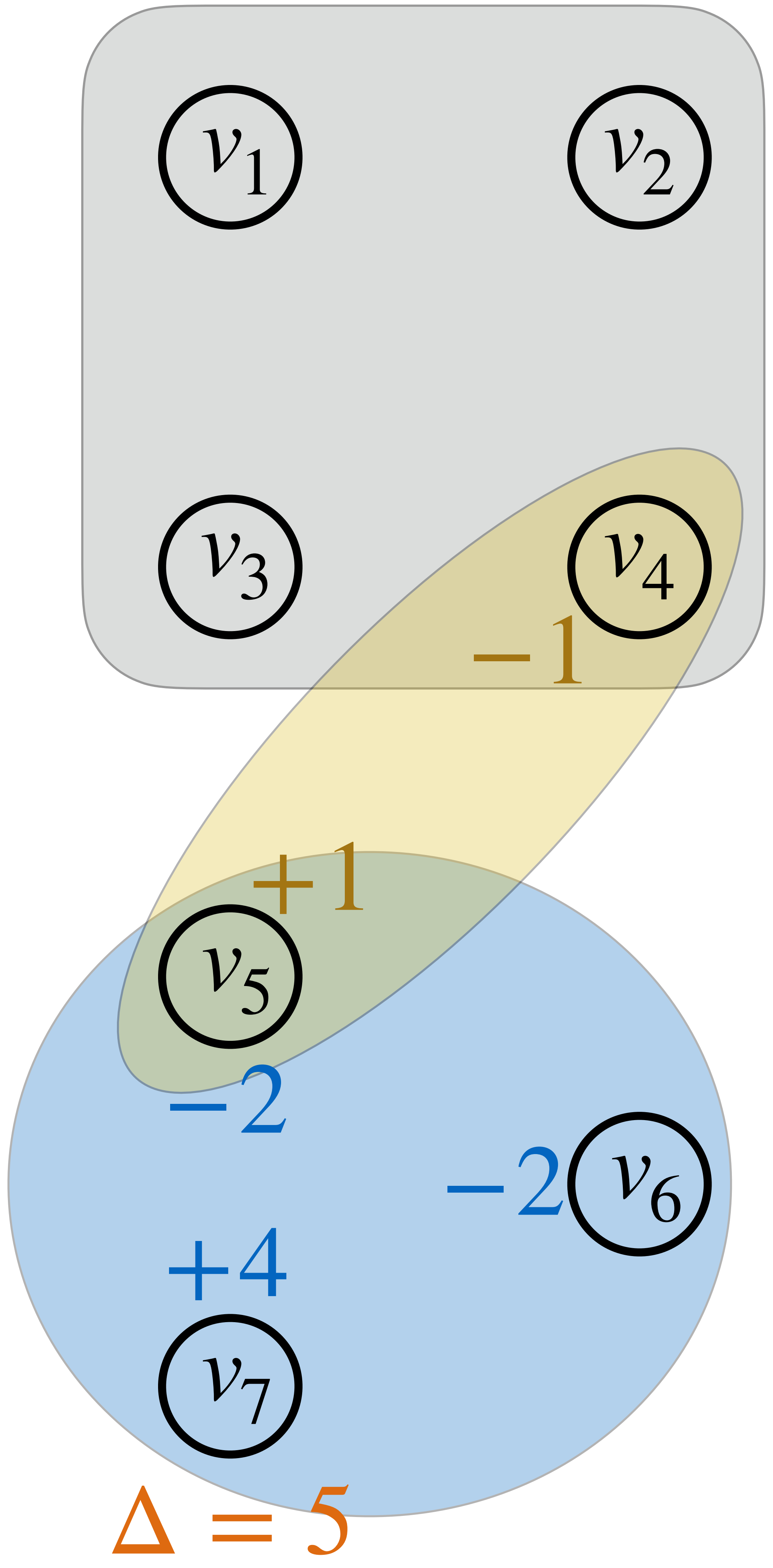
Hyper-Flow Diffusion

- Initial mass Δ on some seed node(s)



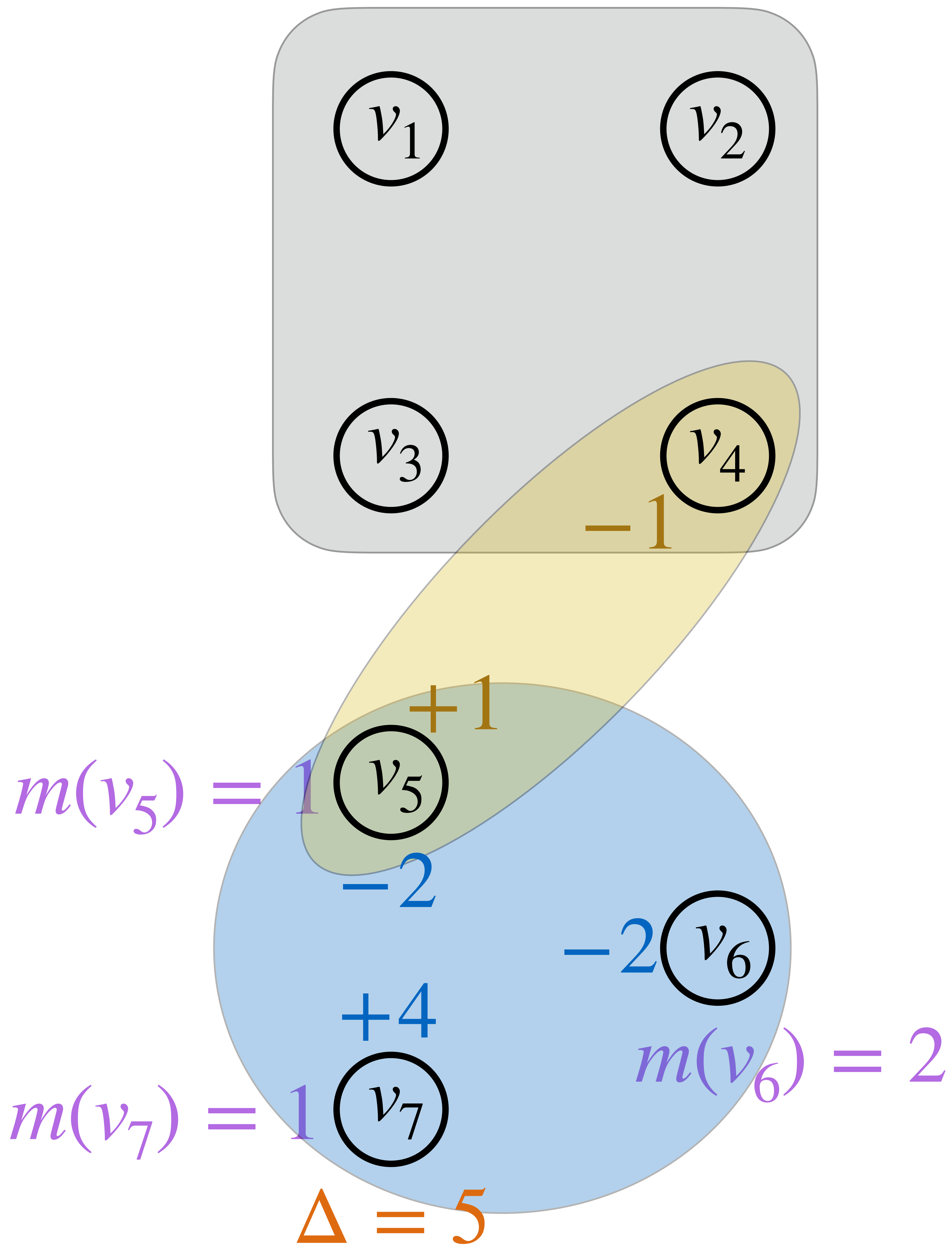
Hyper-Flow Diffusion

- Initial mass Δ on some seed node(s)
- Diffuse mass according to flows over hyperedges



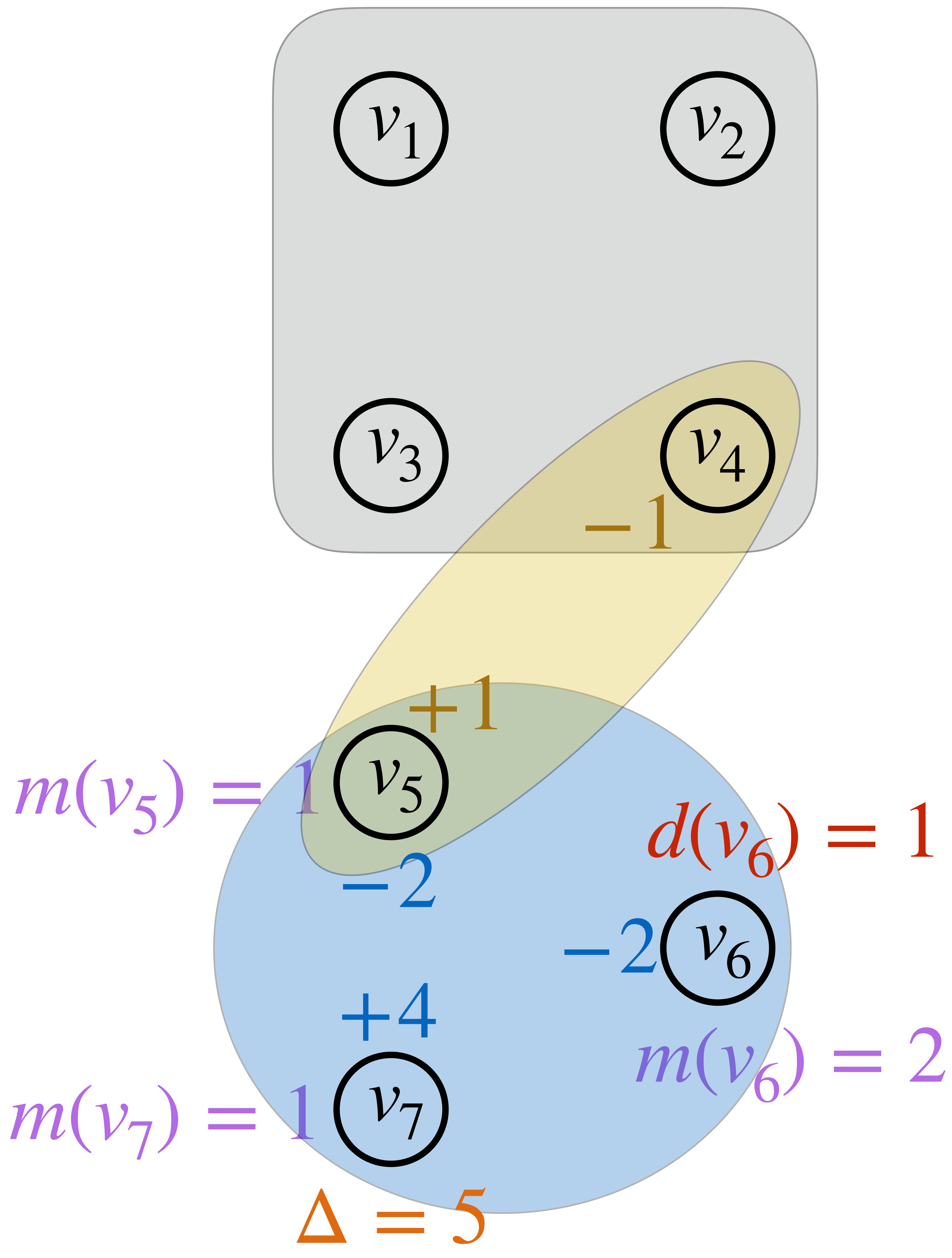
Hyper-Flow Diffusion

- Initial mass Δ on some seed node(s)
- Diffuse mass according to flows over hyperedges
- Leave net mass m on nodes



Hyper-Flow Diffusion

- Initial mass Δ on some seed node(s)
- Diffuse mass according to flows over hyperedges
- Leave net mass m on nodes
- Net mass cannot exceed capacity d

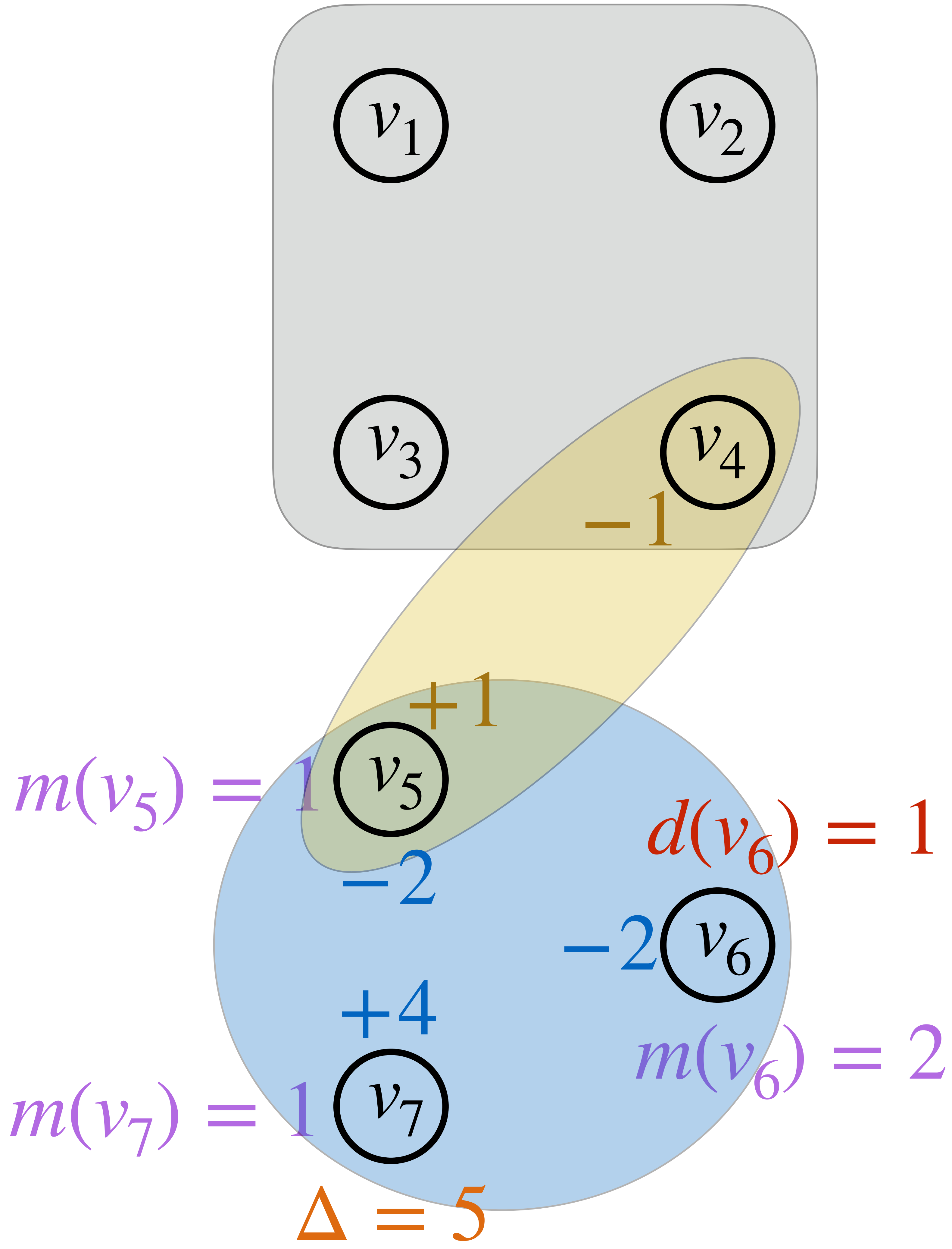


Hyper-Flow Diffusion

- Initial mass Δ on some seed node(s)
- Diffuse mass according to flows over hyperedges
- Leave net mass m on nodes
- Net mass cannot exceed capacity d

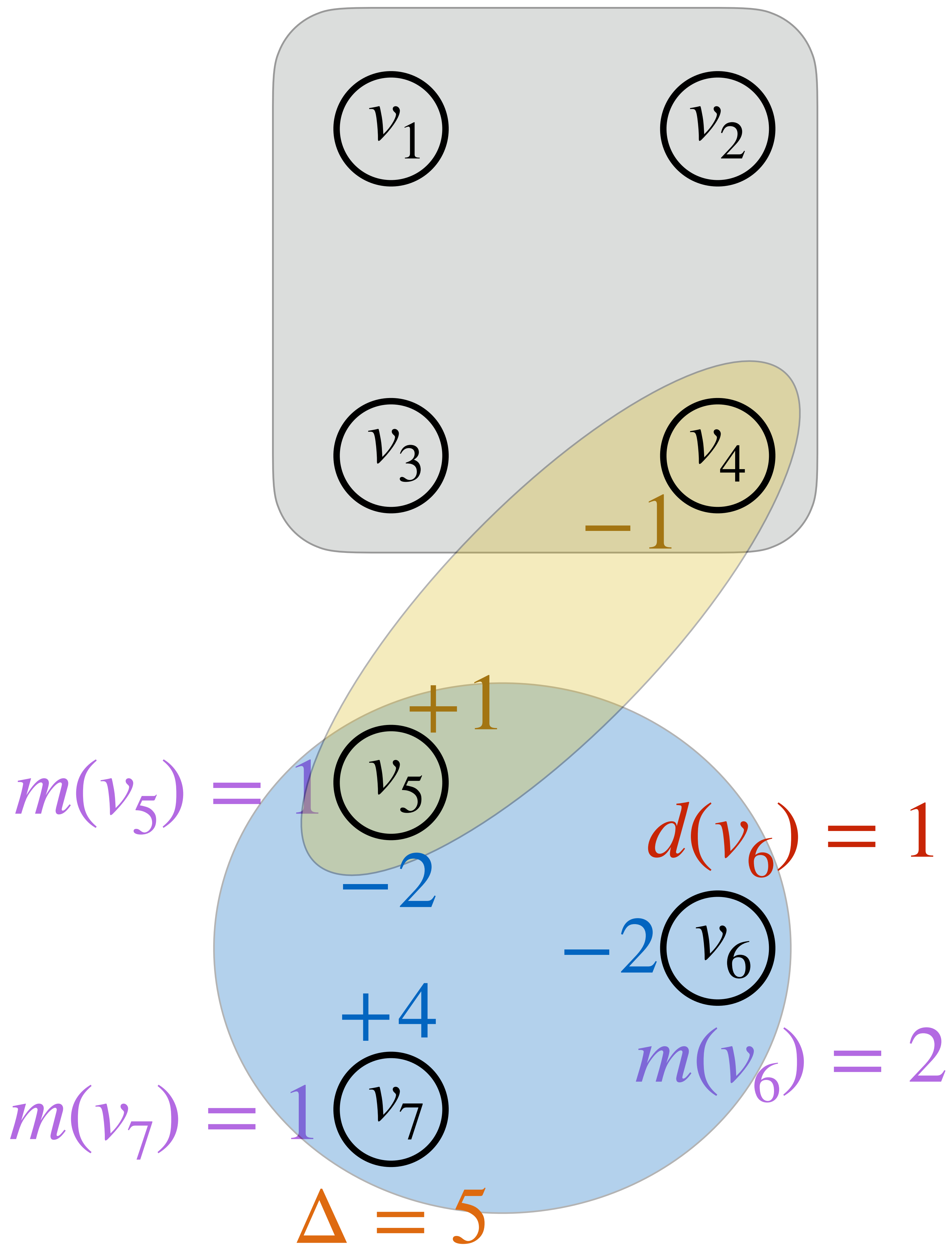
We impose additional constraints so that the flow values respect higher-order relations modelled by the cut-cost function w_e.

Hyper-Flow Diffusion is the diffusion of initial mass according to minimum ℓ_2-norm flow.



Hyper-Flow Diffusion

- Initial mass Δ on some seed node(s)
- Diffuse mass according to flows over hyperedges
- Leave net mass m on nodes
- Net mass cannot exceed capacity d



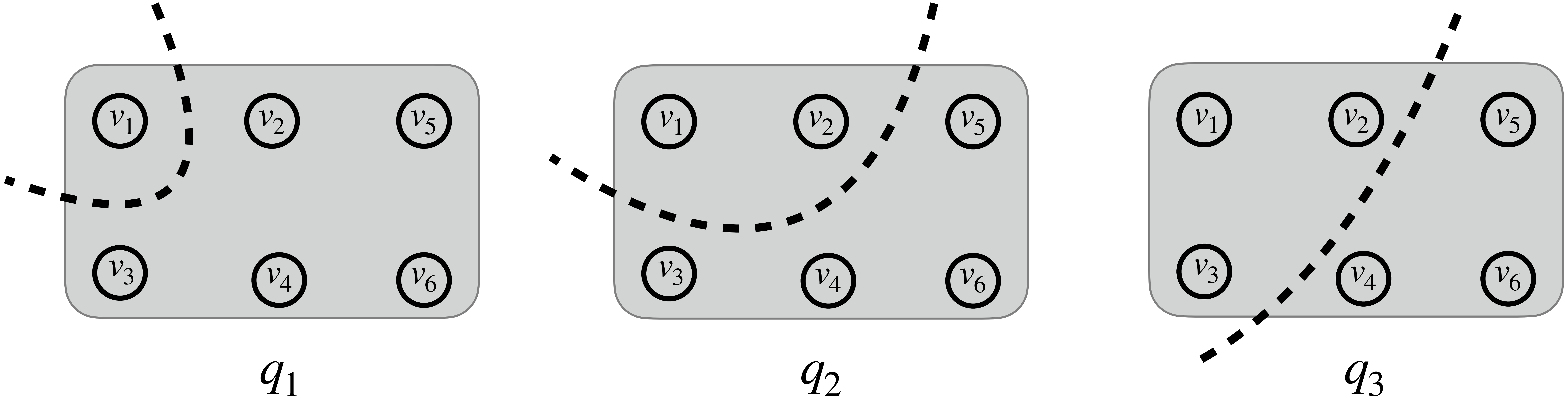
We impose additional constraints so that the flow values respect higher-order relations modelled by the cut-cost function w_e.

Hyper-Flow Diffusion is the diffusion of initial mass according to minimum ℓ_2-norm flow.

We use the excess mass on nodes for node ranking and local clustering

Hyper-Flow Diffusion: empirical results

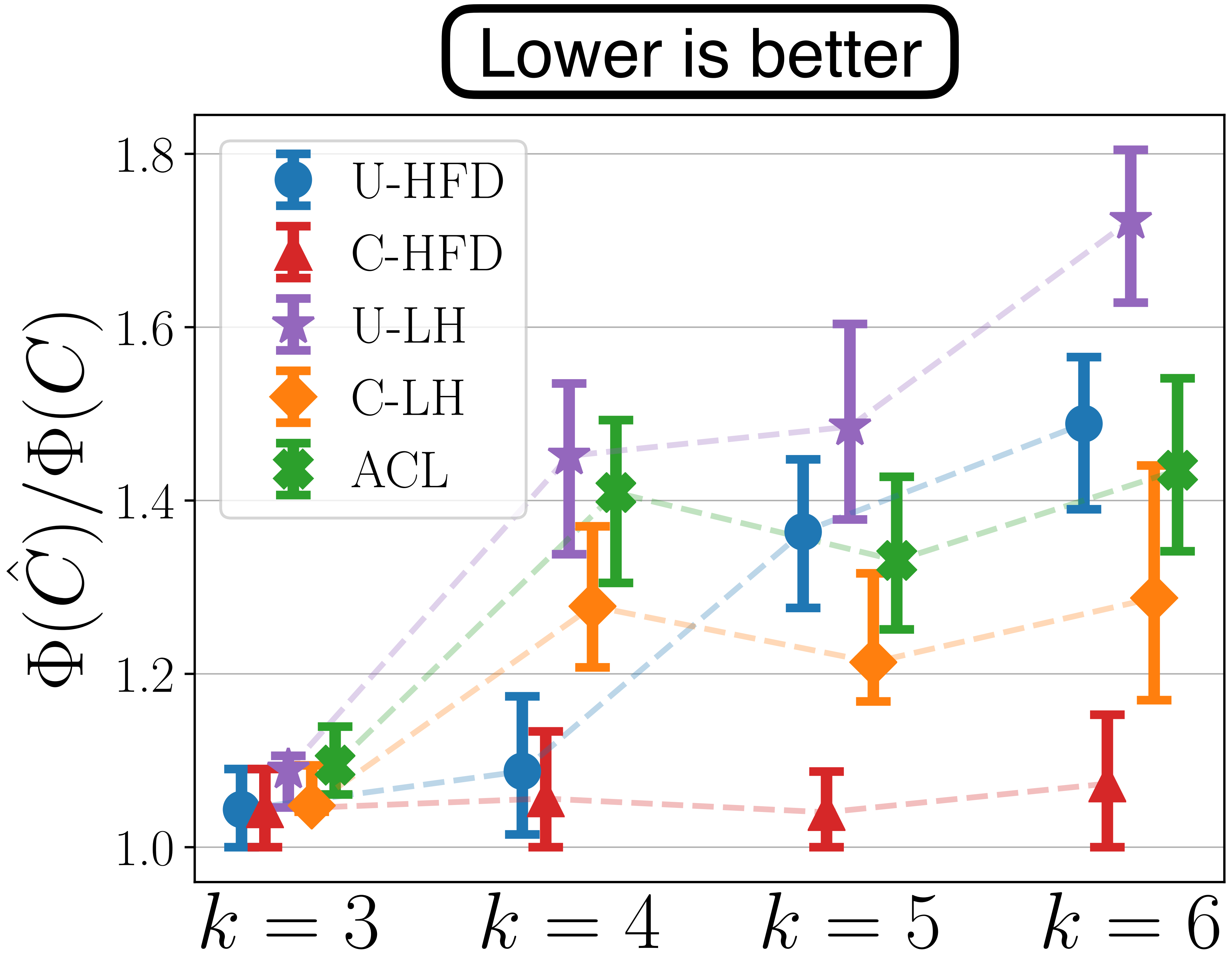
Cardinality-based k-uniform hypergraph stochastic block model:
Boundary hyperedges appear with different probabilities according to the cardinality of hyperedge cut.

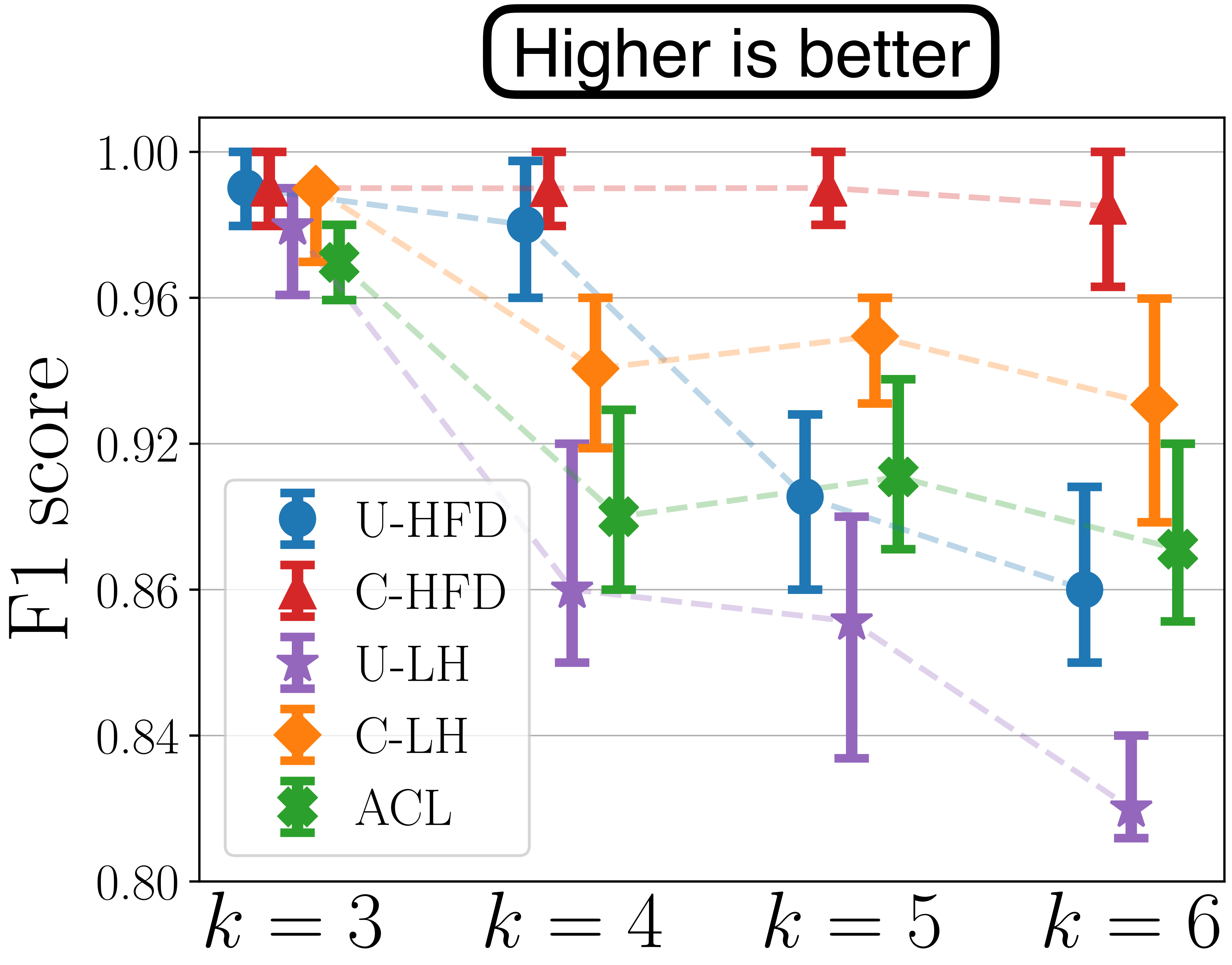


We consider $q_1 \gg q_2 \geq q_3$. Under this generative setting, one should naturally explore cardinality-based cut-cost for clustering.

All our experiments use a single seed node to recover the target

Hyper-Flow Diffusion: empirical results





- LH is a strongly-local hypergraph diffusion method based on graph reduction.
- ACL is a heuristic method that uses PageRank on star expansion.
- HFD is the only method that directly works on original hypergraph.
- U-* means the method uses unit cut-cost; C-* means the method uses cardinality 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

Local clustering on a hypergraph constructed from Amazon product reviews data

Nodes are products

Hyperedges are products reviewed by the same person

Clusters are products belonging to the same product category

			Cluster								
Metric	Seed	Method	1	2	3	12	15	17	18	24	25
Conductance	Single	U-HFD	**0.17**	**0.11**	**0.12**	**0.16**	**0.36**	**0.25**	**0.17**	**0.14**	**0.28**
		U-LH-2.0	0.42	0.50	0.25	0.44	0.74	0.44	0.57	0.58	0.61
		U-LH-1.4	0.33	0.44	0.25	0.36	0.81	0.40	0.51	0.54	0.59
		ACL	0.42	0.50	0.25	0.54	0.77	0.52	0.63	0.68	0.65
	Multiple	U-HFD	**0.05**	**0.10**	**0.12**	**0.13**	**0.20**	**0.16**	**0.14**	**0.11**	**0.32**
		U-LH-2.0	**0.05**	0.15	0.15	0.21	0.45	0.45	0.26	0.18	0.53
		U-LH-1.4	**0.05**	0.13	0.15	0.15	0.35	0.33	0.19	0.14	0.47
		ACL	**0.05**	0.27	0.16	0.27	0.56	0.53	0.33	0.30	0.59
F1 score	Single	U-HFD	**0.45**	**0.09**	**0.65**	**0.92**	0.04	**0.10**	**0.80**	**0.81**	**0.09**
		U-LH-2.0	0.23	0.07	0.23	0.29	**0.05**	0.06	0.21	0.28	0.05
		U-LH-1.4	0.23	**0.09**	0.35	0.40	0.00	0.07	0.31	0.35	0.06
		ACL	0.23	0.07	0.22	0.25	0.04	0.05	0.17	0.20	0.04
	Multiple	U-HFD	0.49	**0.50**	0.69	**0.98**	0.19	**0.36**	**0.91**	**0.89**	**0.33**
		U-LH-2.0	**0.59**	0.42	**0.73**	0.77	0.22	0.25	0.65	0.62	0.17
		U-LH-1.4	0.52	0.45	**0.73**	0.90	**0.27**	0.29	0.79	0.77	0.20
		ACL	**0.59**	0.25	0.70	0.64	0.20	0.19	0.51	0.49	0.14

Hyper-Flow Diffusion: empirical results

Local clustering on a hypergraph constructed from Microsoft academic coauthorthip data

Nodes are papers

Hyperedges are papers having at least a common coauthor

Clusters are papers published at similar venues

Metric	Method	Cluster: Data	ML	TCS	CV
Cond	U-HFD	**0.03**	**0.06**	**0.06**	**0.03**
	U-LH-2.0	0.07	0.09	0.10	0.07
	U-LH-1.4	0.07	0.08	0.09	0.07
	ACL	0.08	0.11	0.11	0.09
F1 score	U-HFD	**0.78**	**0.54**	**0.86**	**0.73**
	U-LH-2.0	0.67	0.46	0.71	0.61
	U-LH-1.4	0.65	0.46	0.59	0.59
	ACL	0.64	0.43	0.70	0.57

Hyper-Flow Diffusion: empirical results

Local clustering on a hypergraph constructed from travel metasearch data (F1 scores)

Nodes are hotel accommodations

Hyperedges are accommodations viewed by the same user in a browsing session

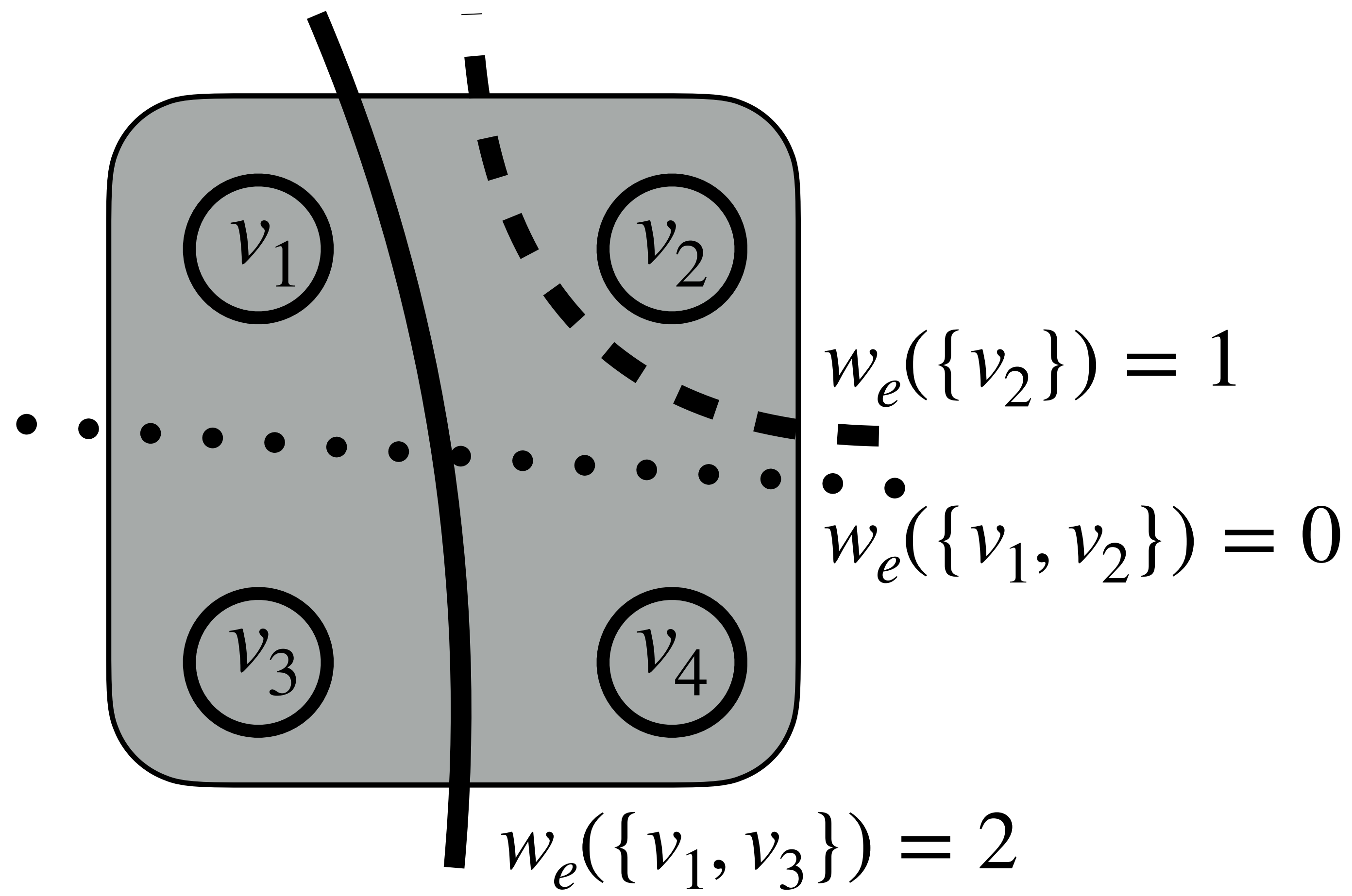
Clusters are accommodations located in the same country/territory

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	Epiphytic Gastropods, Detriti. Gastropods	Meiofauna, Epiphytic Gastropods	0.67	0.47	0.64
S-HFD	Gruiformes, Small Shorebirds	Snook, Mackerel	**0.69**	**0.62**	**0.84**



- S-HFD uses specialized submodular cut-cost shown on the left.
- The example shows that general submodular cut-cost can be necessary.
- HFD is the only local diffusion method that works with general submodular cut-costs.

Hyper-Flow Diffusion: empirical results

For more experiments and details on both synthetic and real datasets:

Please see our preprint **Local Hyper-Flow Diffusion** *on arXiv:2102.07945*

Julia implementation **HFD** on **GitHub**

Thank you!