



Hypergraph p -Laplacians and Scale Spaces

Ariane Fazen^{1,2} · Daniel Tenbrinck¹ · Kseniia Lukin¹ · Martin Burger^{2,3}

Received: 17 October 2023 / Accepted: 29 February 2024 / Published online: 23 April 2024
© The Author(s) 2024

Abstract

The aim of this paper is to revisit the definition of differential operators on hypergraphs, which are a natural extension of graphs in systems based on interactions beyond pairs. In particular, we focus on the definition of Laplacian and p -Laplace operators for oriented and unoriented hypergraphs, their basic properties, variational structure, and their scale spaces. We illustrate that diffusion equations on hypergraphs are possible models for different applications such as information flow on social networks or image processing. Moreover, the spectral analysis and scale spaces induced by these operators provide a potential method to further analyze complex data and their multiscale structure. The quest for spectral analysis and suitable scale spaces on hypergraphs motivates in particular a definition of differential operators with trivial first eigenfunction and thus more interpretable second eigenfunctions. This property is not automatically satisfied in existing definitions of hypergraph p -Laplacians, and we hence provide a novel axiomatic approach that extends previous definitions and can be specialized to satisfy such (or other) desired properties.

Keywords Hypergraphs · PDEs on (hyper)graphs · Diffusion models · Information flow · Hypergraph spectral clustering · Image processing · Denoising · Segmentation

1 Introduction

Methods for image processing, data analysis and simulation of information propagation have strongly benefited from using graph structures in the past, and the modeling with PDEs on graphs including graph p -Laplacians and associated flow became a standard tool for analyzing graph structures and dynamics on such (cf. [6, 15, 16]). Those are carried on in machine learning in the concept of graph neural net-

works, again closely related to models for information flow on networks (cf. [5]). Traditional graphs can, however, capture only *pairwise interactions* of individuals, objects, or pixels in images and thus are unable to directly model group relationships, which are relevant, e.g., in social networks or image patches. In order to mitigate for this problem, we propose to apply a more general structure, namely a **hypergraph** with which it is straightforward to encode group interactions. Here, we adopt the definition of oriented hypergraphs, whose hyperarcs (generalizing edges) can have more than one ingoing and more than one outgoing vertex. For this structure, there is a natural way to define gradients, and we use a scaling which preserves the axiom that the gradient of a constant function on the vertices vanishes. Via a definition of adjoint, we can then obtain a divergence operator and a Laplacian. Additionally, we investigate the case of unoriented hypergraphs, in which so-called hyperedges do not have an orientation, i.e., no distinction between outgoing and ingoing vertices. In contrast to traditional edges in graphs, here the number of vertices per hyperedge is not limited by two. For this type of hypergraph, we introduce two possible Laplacian operators, one which is gradient-based and one via an averaging operator.

The authors acknowledge financial support by the European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 777826 (NoMADS) and the German Science Foundation (DFG) through CRC TR 154 “Mathematical Modelling, Simulation and Optimization Using the Example of Gas Networks”, subproject C06. Part of this work was carried out while M. Burger was with the FAU Erlangen-Nürnberg.

✉ Ariane Fazen
ariane.fazen@desy.de

¹ Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany

² Helmholtz Imaging, Deutsches Elektronen-Synchrotron DESY, Notkestr. 85, 22607 Hamburg, Germany

³ Fachbereich Mathematik, Universität Hamburg, 20146 Hamburg, Germany

1.1 Motivation

The hypergraph structure gives additional flexibility in several applications compared to the pair-based graph structure. An example is the modeling of social phenomena of (fake) news spread, e.g., by connecting one person to all their followers directly and hence representing a community within a social network. One field of study in analyzing information flow in social networks is *opinion formation*, an interesting phenomenon that can be observed for a group of individuals which interact and have complex relationships with each other. Some individuals of the social network, so-called *opinion leaders* or *social media influencers*, with a large group of followers (up to half a billion people) have a strong influence on the opinion of many others and can even make profit by leveraging their impact on large groups of social media users (see, e.g., [17]). Modeling information flow in social networks mathematically is typically performed by using traditional graphs. With such graphs, it is possible to link two social media users with a pairwise connection, if they are online friends or follow each other (see, e.g., [12]). The information flow in the social network can then be modeled in terms of diffusion processes on the graph, e.g., by solving a partial differential equation (PDE) (see, e.g., [1, 4]). However, recent work suggests that interactions beyond pairs are of particular relevance (cf. [18]). Structures reminiscent of a Laplacian on hypergraphs can be found in the model of [14].

A similar question arises in the analysis of community structures, where graph spectral clustering is a standard technique. In order to understand networks including group connections, a more general structure such as hypergraphs seems to be more appropriate. The success of PDE-based methods on graphs motivates a further study on hypergraphs in order to explore the potential of PDEs on such objects. For this sake, we need appropriate definitions of hypergraph gradients and Laplacians, which we revisit in this paper. Moreover, the study of scales on hypergraphs is a relevant topic, which could naturally be defined by the evolution of diffusion type processes we hence study here.

In image processing, the graph structure is potentially limiting, since it merely confines to the comparison of pairs of pixels and their gray (respectively color) values. It may, however, be relevant to make comparison between one pixel and its surrounding pixels without the restriction to all pairs. Another example is nonlocal image processing based on patches consisting of multiple images. Hypergraph p -Laplacians and their associated scale spaces are a promising approach for such.

1.2 Related Work

There already exists extensive literature about traditional graph theory and its application to social networks. In [12], an

overview of social network modeling with traditional graphs is given, including community clustering, similarity analysis, and community-based event detection. It indicates how the versatile structure of a graph can be applied to real-world problems. Arnaboldi et al. [1] introduce the so-called ego network, a graph focusing on one specific social media user in the center and their surrounding concentric layers of followers, sorted hierarchically depending on their contact frequency.

The paper [6] introduces first-order differential operators and a family of p -Laplacian operators for traditional oriented graphs. The proposed partial difference, adjoint, divergence and anisotropic p -Laplacian for traditional graphs are a special case of our vertex gradient, adjoint, divergence, and p -Laplacian operators for hypergraphs, which are introduced in Sect. 3. The theoretical results of [6] are applied for mathematical image analysis, such as filtering, segmentation, clustering, and inpainting, but not for social network modeling.

[9] generalizes the already known p -Laplacian operators for normal graphs to the hypergraph setting and performs spectral analysis with a specific focus on the 1-Laplacian. The spectral properties are then applied to common (hyper)graph problems, for instance vertex partitioning, cuts in graphs, coloring of vertices and hyperarc partitioning, but the paper does not include any numerical experiments or the modeling of social networks with hypergraphs. In comparison, our gradient, adjoint, and p -Laplacian definitions are more general and also have the property of the gradient null space including constant vertex functions. Additionally, they are also more flexible with respect to their adaptability for application tasks.

The use of unoriented hypergraphs to model different sociological phenomena of cliques, such as peer pressure, with consensus models, has been proposed in [14]. Diffusion processes in multi-way interactions with convergence to one united group consensus are modeled with a simple 2-Laplacian inspired by the traditional graph setting. Due to the lack of orientation in the hypergraphs, the described consensus models are not able to capture the effects of a one-sided connection through following someone (e.g., Twitter, Instagram), but only mutual connection through being friends (e.g., Facebook).

Furthermore, [20] uses unoriented hypergraphs in machine learning and shows how hypergraph modeling of data relationships can outperform normal graphs in spectral clustering tasks. Similarly, [11] compares two different algorithms for submodular hypergraph clustering, for not oriented hypergraphs with positive vertex weights and a normalized positive hyperedge weight function, namely the inverse power method (IPM) and the clique expansion method (CEM).

1.3 Main Contributions

The contributions of this paper are manifold. First, we recall the generalized vertex p -Laplacian operators for oriented hypergraphs, which were introduced in the preceding paper [8] and [7]. They generalize the definitions in [9] by including two different vertex weight functions and hyperarc weight functions, respectively. With appropriate choice of these weights, the vertex gradient definition leading to the vertex p -Laplacian fulfills the expected properties of the continuum setting (anti-symmetry and the gradient of a constant function being equal to zero), based on less strict assumptions compared to the implicit gradient of [9].

In order to obtain a meaningful definition of a p -Laplace operator on unoriented hypergraphs as well, we introduce a gradient operator with respect to a single vertex, which follows the idea of the respective operators in the oriented hypergraph case. As an alternative, we also consider an approach via an averaging operator on the unoriented hypergraph, which is, however, confined to the linear case ($p = 2$) as of now. The two different Laplacian operators are subsequently compared in our numerical experiments.

Moreover, we include two possible applications of the corresponding diffusion equations: for the oriented setting we investigate the information flow on networks based on the hypergraph Laplacian and for the unoriented setting we discuss an application to image processing and derive novel scale spaces based on pixel neighborhood comparison, for which our definitions are naturally suited.

2 Mathematical Basics of Hypergraphs

The definition of hypergraphs is a generalization of finite graphs, both in the case of unoriented and oriented hypergraphs, which are based on unoriented and oriented normal graphs, respectively. For a given finite set of vertices $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$, a hypergraph does not only capture pairwise connections between two vertices, but higher-order relationships within any subset of all vertices.

Remark 1 As proposed in [13], we differentiate between oriented and unoriented hypergraphs instead of directed and undirected hypergraphs, because for every oriented hyperarc there is only one orientation but two possible directions: the direction along the orientation and the direction against the orientation.

Definition 1 (Unoriented hypergraph UH) [19] An unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ consists of a finite set of vertices \mathcal{V} , and a set of so-called hyperedges \mathcal{E}_H , with each hyperarc $e_q \in \mathcal{E}_H$ being in the power set of the vertices $2^{|\mathcal{V}|}$ and satisfying $\emptyset \subset e_q \subset \mathcal{V}$ with $2 \leq |e_q| \leq |\mathcal{V}| - 1$.

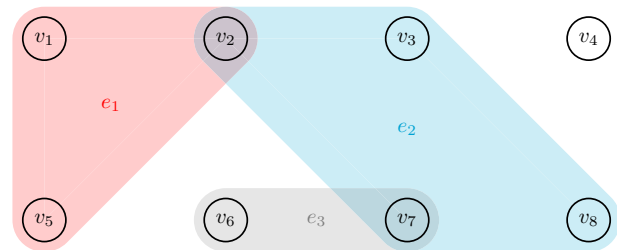
Example 1 (Unoriented hypergraph UH) Given a set of vertices

$$\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$$

and a set of hyperedges

$$\mathcal{E}_H = \{\{v_1, v_2, v_5\}, \{v_2, v_3, v_7, v_8\}, \{v_6, v_7\}\},$$

then the unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ can be visualized in the following way:



Remark 2 For clarity reasons, we assume that each hyperedge in \mathcal{E}_H is unique and hence occurs only once. This implies that the cardinality of the hyperedge set is finite due to set of vertices \mathcal{V} being finite and the number of hyperedges in $UH = (\mathcal{V}, \mathcal{E}_H)$ being limited by $|\mathcal{E}_H| \leq 2^N$.

Assigning either an output or an input orientation to each vertex of a hyperedge results in an oriented version of hyperedges, so-called hyperarcs. Based on this, oriented hypergraphs can be defined.

Definition 2 [Oriented hypergraph OH] [9] An oriented hypergraph $OH = (\mathcal{V}, \mathcal{A}_H)$ consists of a finite set of vertices \mathcal{V} , and a set of so-called hyperarcs \mathcal{A}_H . Each hyperarc $a_q \in \mathcal{A}_H$ contains two disjoint subsets of vertices

$$a_q = (a_q^{out}, a_q^{in}) \quad (1)$$

with $\emptyset \subset a_q^{out}, a_q^{in} \subset \mathcal{V}, a_q^{out} \cap a_q^{in} = \emptyset, a_q^{out}$ being the set of all output vertices and a_q^{in} being the set of all input vertices of the hyperarc a_q .

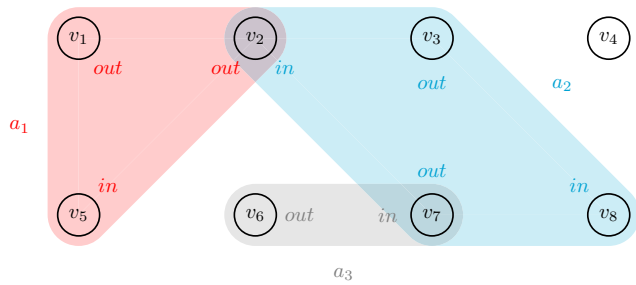
Example 2 [Oriented hypergraph OH] Given a set of vertices

$$\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$$

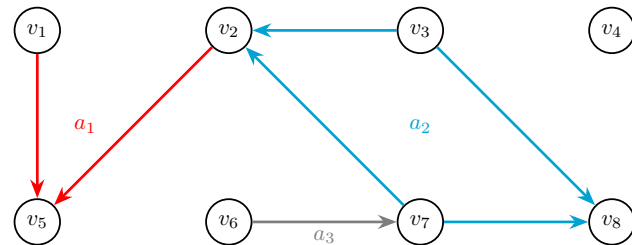
and a set of hyperarcs

$$\mathcal{A}_H = \{(\{v_1, v_2\}, \{v_5\}), (\{v_3, v_7\}, \{v_2, v_8\}), (\{v_6\}, \{v_7\})\},$$

then the oriented hypergraph $OH = (\mathcal{V}, \mathcal{A}_H)$ can be visualized in the following way:



Alternatively, hyperarcs can also be visualized similarly to arcs in normal graphs:



In our numerical experiments, we will use the second visualization option (without color-coding the different hyperarcs) in order to simplify understanding of the links between vertices. Since the underlying oriented hypergraph will have a specific property ($|a_q^{\text{out}}| = 1$), the examples will have a one-to-one correspondence between the "normal graph visualization" and the hypergraph visualization, which would generally not be given without color-coding each hyperarc.

Remark 3 Furthermore, for clarity reasons we assume that each hyperarc in the set of hyperarcs \mathcal{A}_H is unique and thus occurs only once. This automatically implies that the cardinality of the hyperarc set is finite due to set of vertices \mathcal{V} being finite. More precisely the number of hyperarcs in $OH = (\mathcal{V}, \mathcal{A}_H)$ is limited by $|\mathcal{A}_H| \leq N^N$.

We now define different functions on both unoriented and oriented hypergraphs, which are used in Sect. 3 to introduce differential operators inspired by the continuum setting. In order to efficiently denote whether vertex is part of a hyperedge for an unoriented hypergraph and to check if a vertex is part of a hyperarc as an output or an input vertex for an oriented hypergraph, we use different kinds of vertex-hyperedge and vertex-hyperarc characteristic functions.

Definition 3 [Vertex-hyperedge characteristic function δ] For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$, we define the vertex-hyperedge characteristic function δ as:

$$\begin{aligned} \delta : \mathcal{V} \times \mathcal{E}_H &\longrightarrow \{0, 1\} & (v_i, e_q) &\longmapsto \delta(v_i, e_q) \\ &= \begin{cases} 1 & v_i \in e_q \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (2)$$

Definition 4 [Vertex-hyperarc characteristic functions $\delta_{\text{out}}, \delta_{\text{in}}$] For an oriented hypergraph $OH = (\mathcal{V}, \mathcal{A}_H)$, we define the output vertex-hyperarc characteristic function δ_{out} as:

$$\begin{aligned} \delta_{\text{out}} : \mathcal{V} \times \mathcal{A}_H &\longrightarrow \{0, 1\} & (v_i, a_q) &\longmapsto \delta_{\text{out}}(v_i, a_q) \\ &= \begin{cases} 1 & v_i \in a_q^{\text{out}} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (3)$$

Respectively, the input vertex-hyperarc characteristic function δ_{in} is given by:

$$\begin{aligned} \delta_{\text{in}} : \mathcal{V} \times \mathcal{A}_H &\longrightarrow \{0, 1\} & (v_i, a_q) &\longmapsto \delta_{\text{in}}(v_i, a_q) \\ &= \begin{cases} 1 & v_i \in a_q^{\text{in}} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (4)$$

Instead of defining separate δ_{out} and δ_{in} characteristic functions, it would also be possible to define one vertex-hyperarc characteristic function δ_* as:

$$\begin{aligned} \delta_* : \mathcal{V} \times \mathcal{A}_H &\longrightarrow \{-1, 0, 1\} & (v_i, a_q) &\longmapsto \delta_*(v_i, a_q) \\ &= \begin{cases} -1 & v_i \in a_q^{\text{in}} \\ 1 & v_i \in a_q^{\text{out}} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (5)$$

however, this would lead to more complex definitions of the vertex gradient, adjoint, and p -Laplacian operators later on, because it complicates weighing output and input vertices of a hyperarc differently.

Real-valued functions can be defined on the set of vertices \mathcal{V} , the set of hyperedges \mathcal{E}_H , and the set of hyperarcs \mathcal{A}_H in order to link any kind of data to a hypergraph.

Definition 5 [Vertex functions f , and hyperedge or hyperarc functions F] For both an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ and an oriented hypergraph $OH = (\mathcal{V}, \mathcal{A}_H)$, vertex functions are defined on the set of vertices as

$$f : \mathcal{V} \longrightarrow \mathbb{R} \quad v_i \longmapsto f(v_i) \quad (6)$$

with vertex weight functions being defined as

$$w : \mathcal{V} \longrightarrow \mathbb{R}_{>0} \quad v_i \longmapsto w(v_i). \quad (7)$$

For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$, hyperedge functions are defined on the domain of the set of hyperedges as

$$F : \mathcal{E}_H \longrightarrow \mathbb{R} \quad e_q \longmapsto F(e_q) \quad (8)$$

with hyperedge weight functions being defined as

$$W : \mathcal{E}_H \longrightarrow \mathbb{R}_{>0} \quad e_q \longmapsto W(e_q). \quad (9)$$

Similarly, for an oriented hypergraph $OH = (\mathcal{V}, \mathcal{A}_H)$, hyperarc functions are defined on the domain of the set of hyperarcs as

$$F : \mathcal{A}_H \longrightarrow \mathbb{R} \quad a_q \longmapsto F(a_q) \quad (10)$$

with hyperarc weight functions being defined as

$$W : \mathcal{A}_H \longrightarrow \mathbb{R}_{>0} \quad a_q \longmapsto W(a_q). \quad (11)$$

The space of all vertex functions, all hyperedge and all hyperarc functions defined on a given hypergraph can be identified with an N - or an at most N^N -dimensional Hilbert space, respectively.

Definition 6 [Space of vertex functions $\mathcal{H}(\mathcal{V})$, space of hyperedge functions $\mathcal{H}(\mathcal{E}_H)$, and space of hyperarc functions $\mathcal{H}(\mathcal{A}_H)$] For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ and an oriented hypergraph $OH = (\mathcal{V}, \mathcal{A}_H)$, the space of all vertex functions f is given by

$$\mathcal{H}(\mathcal{V}) = \{f \mid f : \mathcal{V} \longrightarrow \mathbb{R}\} \quad (12)$$

where $\mathcal{H}(\mathcal{V})$ with the inner product $\langle f, g \rangle_{\mathcal{H}(\mathcal{V})} = \sum_{v_i \in \mathcal{V}} w_I(v_i)^\alpha f(v_i) g(v_i)$ for any two vertex functions $f, g \in \mathcal{H}(\mathcal{V})$, vertex weight function w_I , and parameter $\alpha \in \mathbb{R}$ is a Hilbert space.

For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$, the space of all hyperedge functions F is defined as

$$\mathcal{H}(\mathcal{E}_H) = \{F \mid F : \mathcal{E}_H \longrightarrow \mathbb{R}\} \quad (13)$$

where $\mathcal{H}(\mathcal{E}_H)$ with the inner product $\langle F, G \rangle_{\mathcal{H}(\mathcal{E}_H)} = \sum_{e_q \in \mathcal{E}_H} W_I(e_q)^\beta F(e_q) G(e_q)$ for any two hyperedge functions $F, G \in \mathcal{H}(\mathcal{E}_H)$, hyperedge weight function W_I , and parameter $\beta \in \mathbb{R}$ constitutes a Hilbert space. In the same manner, the space of all hyperarc functions F for an oriented hypergraph $OH = (\mathcal{V}, \mathcal{A}_H)$ is defined as

$$\mathcal{H}(\mathcal{A}_H) = \{F \mid F : \mathcal{A}_H \longrightarrow \mathbb{R}\} \quad (14)$$

where $\mathcal{H}(\mathcal{A}_H)$ with the product $\langle F, G \rangle_{\mathcal{H}(\mathcal{A}_H)} = \sum_{a_q \in \mathcal{A}_H} W_I(a_q)^\beta F(a_q) G(a_q)$ for any two hyperarc functions $F, G \in \mathcal{H}(\mathcal{A}_H)$, hyperarc weight function W_I , and parameter $\beta \in \mathbb{R}$ defines a Hilbert space.

3 Differential Operators on Hypergraphs

This section introduces first- and higher-order differential operators both for unoriented and for oriented hypergraphs.

3.1 First-order Differential Operators for Oriented Hypergraphs

Utilizing the introduced definitions for hypergraphs, we can now generalize the definitions of the vertex gradient, the vertex adjoint, and the vertex p -Laplacian for normal graphs, which have already been discussed in a simplified form with less weight functions and parameters in [6].

Definition 7 [Vertex gradient operator ∇_v] For an oriented hypergraph $OH = (\mathcal{V}, \mathcal{A}_H)$ with vertex weight functions w_I and w_G , and hyperarc weight function W_G , we define the vertex gradient operator ∇_v with parameters $\alpha, \gamma, \epsilon, \eta \in \mathbb{R}$ as:

$$\begin{aligned} \nabla_v : \mathcal{H}(\mathcal{V}) &\longrightarrow \mathcal{H}(\mathcal{A}_H) \quad f \longmapsto \nabla_v f \\ \nabla_v f : \mathcal{A}_H &\longrightarrow \mathbb{R} \quad a_q \longmapsto \nabla_v f(a_q) \\ &= W_G(a_q)^\gamma \sum_{v_i \in \mathcal{V}} \left(\delta_{\text{in}}(v_i, a_q) \frac{w_I(v_i)^\alpha w_G(v_i)^\epsilon}{|a_q^{\text{in}}|} \right. \\ &\quad \left. - \delta_{\text{out}}(v_i, a_q) \frac{w_I(v_i)^\alpha w_G(v_i)^\eta}{|a_q^{\text{out}}|} \right) f(v_i). \end{aligned} \quad (15)$$

The weight w_I denotes the vertex weight function from the inner product of $\mathcal{H}(\mathcal{V})$ and w_G denotes the vertex weight function, which is introduced with the gradient operator. Using different values for the parameters ϵ and η corresponds to putting different weights on the input and output vertices in the gradient of hyperarc a_q .

The introduced vertex gradient fulfills two expected properties from the continuum setting, namely anti-symmetry and the gradient of a constant function being equal to zero.

Theorem 1 [Vertex gradient operator properties] The vertex gradient ∇_v defined on an oriented hypergraphs $OH = (\mathcal{V}, \mathcal{A}_H)$ with vertex weight functions w_I and w_G , and hyperarc weight function W_G , satisfies the following properties:

(1) **Vanishing gradient of a constant vertex function:** If the condition

$$\begin{aligned} w_I(v_k)^\alpha w_G(v_k)^\epsilon \\ = w_I(v_j)^\alpha w_G(v_j)^\eta \end{aligned}$$

holds for all vertex combinations $v_j, v_k \in \mathcal{V}$ with $v_j \in a_q^{\text{out}}$ and $v_k \in a_q^{\text{in}}$ for a hyperarc $a_q \in \mathcal{A}_H$, then for every constant function f , i.e., $f(v_i) \equiv \bar{f}$ for all vertices $v_i \in \mathcal{V}$, we have $\nabla_v f(a_q) = 0$ for all hyperarcs $a_q \in \mathcal{A}_H$.

(2) **Antisymmetry:** Let $\epsilon = \eta$. Then the identity $\nabla_v f(a_q^{\text{out}}, a_q^{\text{in}}) = -\nabla_v f(a_q^{\text{in}}, a_q^{\text{out}})$ holds for all hyperarcs $a_q \in \mathcal{A}_H$.

Proof See [7] Theorem 9.2 (Vertex gradient operator properties). \square

Let us mention one additional complication compared to the traditional graph case: while it is trivial to see that for a connected graph constant functions are the only elements in the nullspace of the gradient, this is not apparent for hypergraphs.

By computing the adjoint ∇_v^* of the vertex gradient, we can introduce a consistent definition of a divergence operator on hypergraphs in analogy to traditional calculus. Detailed computation based on the relation

$$\langle G, \nabla_v f \rangle_{\mathcal{H}(\mathcal{A}_H)} = \langle f, \nabla_v^* G \rangle_{\mathcal{H}(\mathcal{V})} \quad (16)$$

for all vertex functions $f \in \mathcal{H}(\mathcal{V})$ and all hyperarc functions $G \in \mathcal{H}(\mathcal{A}_H)$ can be found in Theorem 9.9 of [7] (Connection vertex gradient ∇_v and vertex adjoint ∇_v^*).

Definition 8 [Vertex adjoint operator ∇_v^*] For an oriented hypergraph $OH = (\mathcal{V}, \mathcal{A}_H)$ with vertex weight function w_G , and hyperarc weight functions W_I and W_G , the vertex adjoint operator ∇_v^* with parameters $\beta, \gamma, \epsilon, \eta \in \mathbb{R}$ is given by:

$$\begin{aligned} \nabla_v^* : \mathcal{H}(\mathcal{A}_H) &\longrightarrow \mathcal{H}(\mathcal{V}) \quad F \longmapsto \nabla_v^* F \\ \nabla_v^* F : \mathcal{V} &\longrightarrow \mathbb{R} \quad v_i \longmapsto \nabla_v^* F(v_i) \\ &= \sum_{a_q \in \mathcal{A}_H} \left(\delta_{\text{in}}(v_i, a_q) \frac{w_G(v_i)^\epsilon}{|a_q^{\text{in}}|} \right. \\ &\quad \left. - \delta_{\text{out}}(v_i, a_q) \frac{w_G(v_i)^\eta}{|a_q^{\text{out}}|} \right) \\ &\quad W_I(a_q)^\beta W_G(a_q)^\gamma F(a_q). \end{aligned} \quad (17)$$

Definition 9 [Vertex divergence operator div_v] For an oriented hypergraph $OH = (\mathcal{V}, \mathcal{A}_H)$ with vertex weight function w_G , and hyperarc weight functions W_I and W_G , the vertex divergence operator div_v with parameters $\beta, \gamma, \epsilon, \eta \in \mathbb{R}$ is given by:

$$\begin{aligned} \text{div}_v : \mathcal{H}(\mathcal{A}_H) &\longrightarrow \mathcal{H}(\mathcal{V}) \quad F \longmapsto \text{div}_v F \\ \text{div}_v F : \mathcal{V} &\longrightarrow \mathbb{R} \quad v_i \longmapsto \text{div}_v F(v_i) = -\nabla_v^* F(v_i) \\ &= \sum_{a_q \in \mathcal{A}_H} \left(\delta_{\text{out}}(v_i, a_q) \frac{w_G(v_i)^\eta}{|a_q^{\text{out}}|} \right. \\ &\quad \left. - \delta_{\text{in}}(v_i, a_q) \frac{w_G(v_i)^\epsilon}{|a_q^{\text{in}}|} \right) \\ &\quad W_I(a_q)^\beta W_G(a_q)^\gamma F(a_q). \end{aligned}$$

$$\begin{aligned} & - \delta_{\text{in}}(v_i, a_q) \frac{w_G(v_i)^\epsilon}{|a_q^{\text{in}}|} \Big) \\ & W_I(a_q)^\beta W_G(a_q)^\gamma F(a_q). \end{aligned} \quad (18)$$

3.2 p -Laplacian Operators for Oriented Hypergraphs

Based on the previous definitions, we introduce a generalized vertex p -Laplacian inspired by the continuum setting, which implies that for all $p \in (1, \infty)$ and all vertex functions $f \in \mathcal{H}(\mathcal{V})$ it holds true that:

$$\Delta_v^p f = \text{div}_v \left(|\nabla_v f|^{p-2} \nabla_v f \right),$$

where $|\cdot|$ denotes the pointwise absolute value.

Note that from the definition of the divergence as a negative adjoint of the gradient, it becomes clear the oriented hypergraph p -Laplacian is the negative variation of the p -norm of the gradient, which allows to apply the full theory of eigenvalues of p -homogeneous functionals (see [2]). In particular, the oriented hypergraph Laplacian is a negative semidefinite linear operator and has a spectrum on the negative real line.

Definition 10 [Vertex p -Laplacian operator Δ_v^p] For an oriented hypergraph $OH = (\mathcal{V}, \mathcal{A}_H)$ with vertex weight functions w_I and w_G , and with hyperarc weight functions W_I and W_G , the vertex p -Laplacian operator Δ_v^p with parameters $\alpha, \beta, \gamma, \epsilon, \eta \in \mathbb{R}$ is given by:

$$\begin{aligned} \Delta_v^p : \mathcal{H}(\mathcal{V}) &\longrightarrow \mathcal{H}(\mathcal{V}) \quad f \longmapsto \Delta_v^p f \\ f &\longmapsto \Delta_v^p f : \mathcal{V} \longrightarrow \mathbb{R} \quad v_i \longmapsto \Delta_v^p f(v_i) \\ &= \sum_{a_q \in \mathcal{A}_H} \left(\delta_{\text{out}}(v_i, a_q) \frac{w_G(v_i)^\eta}{|a_q^{\text{out}}|} \right. \\ &\quad \left. - \delta_{\text{in}}(v_i, a_q) \frac{w_G(v_i)^\epsilon}{|a_q^{\text{in}}|} \right) W_I(a_q)^\beta W_G(a_q)^{p\gamma} \\ &\quad \times \left| \sum_{v_j \in \mathcal{V}} \left(\delta_{\text{in}}(v_j, a_q) \frac{w_I(v_j)^\alpha w_G(v_j)^\epsilon}{|a_q^{\text{in}}|} \right. \right. \\ &\quad \left. \left. - \delta_{\text{out}}(v_j, a_q) \frac{w_I(v_j)^\alpha w_G(v_j)^\eta}{|a_q^{\text{out}}|} \right) f(v_j) \right|^{p-2} \\ &\quad \times \sum_{v_k \in \mathcal{V}} \left(\delta_{\text{in}}(v_k, a_q) \frac{w_I(v_k)^\alpha w_G(v_k)^\epsilon}{|a_q^{\text{in}}|} \right. \\ &\quad \left. - \delta_{\text{out}}(v_k, a_q) \frac{w_I(v_k)^\alpha w_G(v_k)^\eta}{|a_q^{\text{out}}|} \right) f(v_k). \end{aligned} \quad (19)$$

The following theorem states that the vertex p -Laplacian is well-defined.

Theorem 2 [Connection vertex gradient ∇_v , vertex divergence div_v , and vertex p -Laplacian Δ_v^p] For an oriented hypergraph $OH = (\mathcal{V}, \mathcal{A}_H)$ with vertex weight functions w_I and w_G , and hyperarc weight functions W_I and W_G , the vertex p -Laplacian Δ_v^p fulfills the equality

$$\Delta_v^p f = \text{div}_v \left(|\nabla_v f|^{p-2} \nabla_v f \right) \quad (20)$$

for all vertex functions $f \in \mathcal{H}(\mathcal{V})$.

Proof See Theorem 10.13 in [7] (Connection vertex divergence div_v , vertex gradient ∇_v , and vertex p -Laplacian Δ_v^p). \square

Moreover, our vertex p -Laplacian definition is a valid generalization of the definition introduced in [9].

Remark 4 [Parameter choice for the vertex p -Laplacian operator] The simplified definition of the vertex p -Laplacian introduced in [9] for any vertex function $f \in \mathcal{H}(\mathcal{V})$ and for any vertex $v_i \in \mathcal{V}$ can be written in our notation as:

$$\begin{aligned} \Delta_p f(v_i) &= \frac{1}{\deg(v_i)} \sum_{\substack{a_q \in \mathcal{A}_H: \delta_{\text{out}}(v_i, a_q)=1 \\ \text{or } \delta_{\text{in}}(v_i, a_q)=1}} \\ &\times \left| \sum_{v_j \in a_q^{\text{in}}} f(v_j) - \sum_{v_j \in a_q^{\text{out}}} f(v_j) \right|^{p-2} \\ &\times \left(\sum_{v_k \in \mathcal{V}} (\delta_{\text{out}}(v_i, a_q) \delta_{\text{out}}(v_k, a_q) \right. \\ &\quad \left. + \delta_{\text{in}}(v_i, a_q) \delta_{\text{in}}(v_k, a_q)) f(v_k) \right. \\ &\quad \left. - \sum_{v_k \in \mathcal{V}} (\delta_{\text{out}}(v_i, a_q) \delta_{\text{in}}(v_k, a_q) \right. \\ &\quad \left. + \delta_{\text{in}}(v_i, a_q) \delta_{\text{out}}(v_k, a_q)) f(v_k) \right). \quad (21) \end{aligned}$$

The factor $(\delta_{\text{out}}(v_i, a_q) \delta_{\text{out}}(v_k, a_q) + \delta_{\text{in}}(v_i, a_q) \delta_{\text{in}}(v_k, a_q))$ is always equal to zero, unless $v_i, v_k \in a_q^{\text{out}}$ or $v_i, v_k \in a_q^{\text{in}}$, which means that the vertices v_i and v_k are co-oriented. Similarly, the factor $(\delta_{\text{out}}(v_i, a_q) \delta_{\text{in}}(v_k, a_q) + \delta_{\text{in}}(v_i, a_q) \delta_{\text{out}}(v_k, a_q))$ ensures to only consider vertices $v_k \in \mathcal{V}$ which are anti-oriented compared to vertex v_i and hence either $v_i \in a_q^{\text{out}}, v_k \in a_q^{\text{in}}$ or $v_i \in a_q^{\text{in}}, v_k \in a_q^{\text{out}}$.

Thus, choosing the parameters of the vertex p -Laplacian Δ_v^p as $\alpha = 0, \beta = 0, \gamma = 0, \epsilon = 0$ and $\eta = 0$ together with excluding the $\frac{1}{|a_q^{\text{out}}|}$ and $\frac{1}{|a_q^{\text{in}}|}$ multiplicative factors and including a new $-\frac{1}{\deg(v_i)}$ factor in the vertex adjoint and the vertex

divergence, results in the simplified vertex p -Laplacian introduced in [9].

Moreover, applying these parameter choices to the vertex gradient, the vertex adjoint, and the vertex divergence leads to the following definitions for all vertex functions $f \in \mathcal{H}(\mathcal{V})$, all hyperarc functions $F \in \mathcal{H}(\mathcal{A}_H)$, for all hyperarcs $a_q \in \mathcal{A}_H$ and all vertices $v_i \in \mathcal{V}$:

$$\nabla_v f(a_q) = \sum_{v_i \in \mathcal{V}} (\delta_{\text{in}}(v_i, a_q) - \delta_{\text{out}}(v_i, a_q)) f(v_i)$$

$$\nabla_v^* F(v_i) = -\frac{1}{\deg(v_i)} \sum_{a_q \in \mathcal{A}_H} (\delta_{\text{in}}(v_i, a_q) - \delta_{\text{out}}(v_i, a_q)) F(a_q)$$

$$\text{div}_v(F)(v_i) = -\frac{1}{\deg(v_i)} \sum_{a_q \in \mathcal{A}_H} (\delta_{\text{out}}(v_i, a_q) - \delta_{\text{in}}(v_i, a_q)) F(a_q)$$

Proof See Theorem 10.12 in [7] (Parameter choice for the vertex p -Laplacian operator). \square

Based on our axiomatic definition of the p -Laplacian via a gradient and adjoint divergence, it is straightforward to verify its variational structure:

Theorem 3 [p -Laplacian energy and derivatives] For an oriented hypergraph $OH = (\mathcal{V}, \mathcal{A}_H)$, the negative hypergraph p -Laplacian for $p \in (1, \infty)$ is the first variation of the associated p -Dirichlet energy

$$E_p[f] := \sum_{v_i \in \mathcal{V}} |\nabla_v^p f(v_i)|, \quad (22)$$

i.e., for every vertex function $f \in \mathcal{H}(\mathcal{V})$ we have

$$-\Delta_v^p f = E'_p[f]. \quad (23)$$

3.3 First-order Differential Operators for Unoriented Hypergraphs

In order to retrieve a meaningful definition of a gradient also in the case of an unoriented hypergraph, where vertices in a hyperedge cannot be separated into output and input vertices, we appoint for each hyperedge $e_q \in \mathcal{E}_H$ a specific vertex $v_{\tilde{q}} := v_i \in e_q$, which all other vertices in the hyperedge $v_j \in e_q \setminus \{v_{\tilde{q}}\}$ are compared to.

Remark 5 If it is not clear how to choose a suitable special vertex $v_{\tilde{q}}$ for each hyperedge $e_q \in \mathcal{E}_H$ based on the application, then it is also possible to include each hyperedge e_q exactly $|e_q|$ times in the set of hyperedges \mathcal{E}_H , where each version of the hyperedge has a different special vertex $v_i \in e_q$ (however, note that this requires paying particular attention to

notation as mentioned in Remark 2). This means that we are able to generate an oriented hypergraph out of an unoriented one as follows: for each hyperedge $e_q \in \mathcal{E}_H$ in the unoriented hypergraph, we create $|e_q|$ hyperarcs for the oriented hypergraph with the same vertices as e_q . Each hyperarc has one output and $|e_q| - 1$ input vertices and each vertex $v_i \in e_q$ is an output vertex in exactly one newly created hyperarc. The associated vertex gradient, adjoint, and p -Laplacian operators for the oriented hypergraph then follow the respective definitions of the unoriented hypergraph case.

Before defining new differential operators for unoriented hypergraphs, it is necessary to introduce a new vertex-hyperedge characteristic function.

Definition 11 [Vertex-hyperedge characteristic function $\tilde{\delta}$] For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$, we define the vertex-hyperedge characteristic function $\tilde{\delta}$ as

$$\begin{aligned} \tilde{\delta} : \mathcal{V} \times \mathcal{E}_H &\longrightarrow \{0, 1\} \\ (v_i, e_q) &\longmapsto \tilde{\delta}(v_i, e_q) = \begin{cases} 1 & v_i = v_{\tilde{q}} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (24)$$

which indicates if vertex $v_i \in \mathcal{V}$ is the special vertex $v_{\tilde{q}}$ of hyperedge $e_q \in \mathcal{E}_H$. Furthermore, the following connection to the vertex-hyperedge function δ holds true for all vertices $v_i \in \mathcal{V}$ and all hyperedges $e_q \in \mathcal{E}_H$:

$$\tilde{\delta}(v_i, e_q) = 1 \implies \delta(v_i, e_q) = 1. \quad (25)$$

The vertex gradient operator for unoriented hypergraphs is defined with the same weight functions and parameters as the definition in the oriented case.

Definition 12 [Vertex gradient operator ∇_v] For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ with vertex weight functions w_I and w_G , and hyperedge weight function W_G , the vertex gradient operator ∇_v with parameters $\alpha, \gamma, \epsilon, \eta \in \mathbb{R}$ is given as:

$$\begin{aligned} \nabla_v : \mathcal{H}(\mathcal{V}) &\longrightarrow \mathcal{H}(\mathcal{E}_H) \\ f &\longmapsto \nabla_v f \\ \nabla_v f : \mathcal{E}_H &\longrightarrow \mathbb{R} \quad e_q \longmapsto \nabla_v f(e_q) \end{aligned}$$

with

$$\begin{aligned} \nabla_v f(e_q) &= W_G(e_q)^\gamma \left(\sum_{v_i \in \mathcal{V}} \delta(v_i, e_q) \left(w_I(v_i)^\alpha \right. \right. \\ &\quad \times w_G(v_i)^\epsilon f(v_i) - w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \Big) \Big). \end{aligned}$$

The above gradient can be rewritten as:

$$\begin{aligned} \nabla_v f(e_q) &= W_G(e_q)^\gamma \left(\left(\sum_{v_i \in \mathcal{V}} \delta(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) \right) \right. \\ &\quad \left. - |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \right). \end{aligned} \quad (26)$$

The vertex gradient for unoriented hypergraphs also fulfills the expected property of a constant vertex function $f \in \mathcal{H}(\mathcal{V})$ resulting in a vanishing gradient.

Theorem 4 [Vertex gradient operator properties] The vertex gradient ∇_v of an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ with vertex weight functions w_I and w_G , and hyperedge weight function W_G , satisfies the following property: if the vertex weights suffice the condition

$$w_I(v_i)^\alpha w_G(v_i)^\epsilon = w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta$$

for all vertex-hyperedge combinations $v_i \in e_q$ and $e_q \in \mathcal{E}_H$, then for every constant function $f \in \mathcal{H}(\mathcal{V})$, i.e., $f(v_i) \equiv \bar{f}$ for all vertices $v_i \in \mathcal{V}$, we get $\nabla_v f(e_q) = 0$ for all hyperedges $e_q \in \mathcal{A}_H$.

Proof Given a constant vertex function $f \in \mathcal{H}(\mathcal{V})$ on an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ with vertex weight functions w_I and w_G , and hyperedge weight function W_G , then it holds true that: the property $w_I(v_i)^\alpha w_G(v_i)^\epsilon = w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta$ for all vertex-hyperedge combinations $v_i \in e_q$ and $e_q \in \mathcal{E}_H$ implies that for each hyperedge e_q there exists a constant $w_{e_q} \in \mathbb{R}_{>0}$ such that

$$w_I(v_i)^\alpha w_G(v_i)^\epsilon = w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta =: w_{e_q}$$

for all vertices $v_i \in e_q$. Thus, together with the property $f(v_i) \equiv \bar{f} \in \mathbb{R}$ for all vertices $v_i \in \mathcal{V}$, this yields for every hyperedge $e_q \in \mathcal{E}_H$:

$$\begin{aligned} \nabla_v f(e_q) &= W_G(e_q)^\gamma \left(\sum_{v_i \in \mathcal{V}} \delta(v_i, e_q) \left(w_I(v_i)^\alpha w_G(v_i)^\epsilon \bar{f} \right. \right. \\ &\quad \left. \left. - w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta \bar{f} \right) \right) \\ &= W_G(e_q)^\gamma \left(\sum_{v_i \in \mathcal{V}} \delta(v_i, e_q) \left(w_I(v_i)^\alpha w_G(v_i)^\epsilon \right. \right. \\ &\quad \left. \left. - w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta \right) \bar{f} \right) \\ &= W_G(e_q)^\gamma \left(\sum_{v_i \in \mathcal{V}} \delta(v_i, e_q) (w_{e_q} - w_{e_q}) \bar{f} \right) \end{aligned}$$

$$\begin{aligned}
&= W_G (e_q)^\gamma |e_q| (w_{e_q} - w_{e_q}) \bar{f} \\
&= W_G (e_q)^\gamma |e_q| \cdot 0 \cdot \bar{f} = 0
\end{aligned}$$

Where the last equality is feasible due to the hyperedge weight function W_G and the vertex function \bar{f} being real functions and the number of vertices in every hyperedge $|e_q|$ being finite. \square

Based on the connections in the continuum setting

$$\begin{aligned}
\langle G, \nabla_v f \rangle_{\mathcal{H}(\mathcal{E}_H)} &= \langle f, \nabla_v^* G \rangle_{\mathcal{H}(\mathcal{V})} \\
\operatorname{div}_v F &= -\nabla_v^* F
\end{aligned}$$

for all vertex functions $f \in \mathcal{H}(\mathcal{V})$ and all hyperedge functions $F, G \in \mathcal{H}(\mathcal{E}_H)$, we define the vertex adjoint and vertex gradient operators for unoriented hypergraphs.

Definition 13 [Vertex adjoint operator ∇_v^*] For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ with vertex weight function w_G , and hyperedge weight functions W_I and W_G , the vertex adjoint operator ∇_v^* with parameters $\beta, \gamma, \epsilon, \eta \in \mathbb{R}$ is given by:

$$\begin{aligned}
\nabla_v^* : \mathcal{H}(\mathcal{E}_H) &\longrightarrow \mathcal{H}(\mathcal{V}) \quad F \longmapsto \nabla_v^* F \\
\nabla_v^* F : \mathcal{V} &\longrightarrow \mathbb{R} \quad v_i \longmapsto \nabla_v^* F(v_i) = \\
&\sum_{e_q \in \mathcal{E}_H} (\delta(v_i, e_q) w_G(v_i)^\epsilon \\
&\quad - \tilde{\delta}(v_i, e_q) |e_q| \\
&\quad w_G(v_i)^\eta) W_I(e_q)^\beta W_G(e_q)^\gamma F(e_q). \quad (27)
\end{aligned}$$

Theorem 5 [Connection vertex gradient ∇_v and vertex adjoint ∇_v^*] For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ with vertex weight functions w_I and w_G , and hyperedge weight functions W_I and W_G , the vertex gradient ∇_v and the vertex adjoint ∇_v^* fulfill the equality

$$\langle G, \nabla_v f \rangle_{\mathcal{H}(\mathcal{E}_H)} = \langle f, \nabla_v^* G \rangle_{\mathcal{H}(\mathcal{V})} \quad (28)$$

for all vertex functions $f \in \mathcal{H}(\mathcal{V})$ and all hyperedge functions $G \in \mathcal{H}(\mathcal{E}_H)$.

Proof For the sake of clarity, the proof is given in the appendix. \square

As in the case of the oriented hypergraph, we define the vertex divergence operator based on the vertex adjoint operator.

Definition 14 [Vertex divergence operator div_v] For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ with vertex weight function w_G , and hyperedge weight functions W_I and W_G , the vertex divergence operator div_v with parameters $\beta, \gamma, \epsilon, \eta \in \mathbb{R}$ is given by:

$$\begin{aligned}
\operatorname{div}_v : \mathcal{H}(\mathcal{E}_H) &\longrightarrow \mathcal{H}(\mathcal{V}) \quad F \longmapsto \operatorname{div}_v F \\
\operatorname{div}_v F : \mathcal{V} &\longrightarrow \mathbb{R} \quad v_i \longmapsto \\
\operatorname{div}_v F(v_i) &= -\nabla_v^* F(v_i) = \sum_{e_q \in \mathcal{E}_H} \left(\tilde{\delta}(v_i, e_q) |e_q| w_G(v_i)^\eta \right. \\
&\quad \left. - \delta(v_i, e_q) w_G(v_i)^\epsilon \right) W_I(e_q)^\beta W_G(e_q)^\gamma F(e_q). \quad (29)
\end{aligned}$$

3.4 p -Laplacian Operators for Unoriented Hypergraphs

Analogously to the case of the oriented hypergraph, in this subsection we present a definition for the vertex p -Laplacian based on the vertex gradient and vertex divergence. The vertex Laplacian we obtain from a perspective of averaging can be found in the next subsection.

Definition 15 [Vertex p -Laplacian operator Δ_v^p] For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ with vertex weight functions w_I and w_G , and hyperedge weight functions W_I and W_G , the vertex p -Laplacian operator Δ_v^p with parameters $\alpha, \beta, \gamma, \epsilon, \eta \in \mathbb{R}$ is given by:

$$\begin{aligned}
\Delta_v^p : \mathcal{H}(\mathcal{V}) &\longrightarrow \mathcal{H}(\mathcal{V}) \quad f \longmapsto \Delta_v^p f \quad \Delta_v^p f : \mathcal{V} \longrightarrow \mathbb{R} \\
v_i &\longmapsto \Delta_v^p f(v_i) \\
&= \sum_{e_q \in \mathcal{E}_H} \left(\tilde{\delta}(v_i, e_q) |e_q| w_G(v_i)^\eta \right. \\
&\quad \left. - \delta(v_i, e_q) w_G(v_i)^\epsilon \right) W_I(e_q)^\beta W_G(e_q)^{p\gamma} \\
&\quad \times \left| \left(\sum_{v_j \in \mathcal{V}} \delta(v_j, e_q) w_I(v_j)^\alpha \right. \right. \\
&\quad \left. \left. w_G(v_j)^\epsilon f(v_j) - |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \right) \right|^{p-2} \\
&\quad \times \left(\left(\sum_{v_k \in \mathcal{V}} \delta(v_k, e_q) w_I(v_k)^\alpha w_G(v_k)^\epsilon f(v_k) \right) \right. \\
&\quad \left. - |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \right). \quad (30)
\end{aligned}$$

Theorem 6 [Connection vertex gradient ∇_v , vertex divergence div_v , and vertex p -Laplacian Δ_v^p] For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ with vertex weight functions w_I and w_G , and hyperedge weight functions W_I and W_G , the presented vertex p -Laplacian Δ_v^p fulfills the equality

$$\Delta_v^p f = \operatorname{div}_v \left(|\nabla_v f|^{p-2} \nabla_v f \right) \quad (31)$$

for all vertex functions $f \in \mathcal{H}(\mathcal{V})$.

Proof Given an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ with vertex weight functions w_I and w_G , and hyperedge weight functions W_I and W_G , and a vertex function $f \in \mathcal{H}(\mathcal{V})$, then the definitions of the vertex divergence operator div_v and the vertex gradient operator ∇_v lead to the following for all vertices $v_i \in \mathcal{V}$:

$$\begin{aligned} \text{div}_v \left(|\nabla_v f|^{p-2} \nabla_v f \right) (v_i) &= \sum_{e_q \in \mathcal{E}_H} \left(\tilde{\delta}(v_i, e_q) |e_q| w_G(v_i)^\eta \right. \\ &\quad \left. - \delta(v_i, e_q) w_G(v_i)^\epsilon \right) W_I(e_q)^\beta W_G(e_q)^\gamma \\ &\quad |\nabla_v f(e_q)|^{p-2} \nabla_v f(e_q) \\ &= \sum_{e_q \in \mathcal{E}_H} \left(\tilde{\delta}(v_i, e_q) |e_q| w_G(v_i)^\eta \right. \\ &\quad \left. - \delta(v_i, e_q) w_G(v_i)^\epsilon \right) W_I(e_q)^\beta W_G(e_q)^\gamma \\ &\quad \left| W_G(e_q)^\gamma \left(\left(\sum_{v_j \in \mathcal{V}} \delta(v_j, e_q) w_I(v_j)^\alpha w_G(v_j)^\epsilon f(v_j) \right) \right. \right. \\ &\quad \left. \left. - |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \right) \right|^{p-2} \\ &\quad W_G(e_q)^\gamma \left(\left(\sum_{v_k \in \mathcal{V}} \delta(v_k, e_q) w_I(v_k)^\alpha w_G(v_k)^\epsilon f(v_k) \right) \right. \\ &\quad \left. - |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \right) \end{aligned}$$

Since the hyperedge weight function W_G maps to positive values, the following equality holds true and leads to the vertex p -Laplacian definition for unoriented hypergraphs:

$$\begin{aligned} &= \sum_{e_q \in \mathcal{E}_H} \left(\tilde{\delta}(v_i, e_q) |e_q| w_G(v_i)^\eta \right. \\ &\quad \left. - \delta(v_i, e_q) w_G(v_i)^\epsilon \right) W_I(e_q)^\beta W_G(e_q)^{\gamma+\gamma(p-2)+\gamma} \\ &\quad \left| \left(\sum_{v_j \in \mathcal{V}} \delta(v_j, e_q) w_I(v_j)^\alpha w_G(v_j)^\epsilon f(v_j) \right) \right. \\ &\quad \left. - |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \right|^{p-2} \\ &\quad \left(\left(\sum_{v_k \in \mathcal{V}} \delta(v_k, e_q) w_I(v_k)^\alpha w_G(v_k)^\epsilon f(v_k) \right) \right. \\ &\quad \left. - |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \right) \end{aligned}$$

$$\begin{aligned} &= \sum_{e_q \in \mathcal{E}_H} \left(\tilde{\delta}(v_i, e_q) |e_q| w_G(v_i)^\eta \right. \\ &\quad \left. - \delta(v_i, e_q) w_G(v_i)^\epsilon \right) W_I(e_q)^\beta W_G(e_q)^{p\gamma} \\ &\quad \left| \left(\sum_{v_j \in \mathcal{V}} \delta(v_j, e_q) w_I(v_j)^\alpha w_G(v_j)^\epsilon f(v_j) \right) \right. \\ &\quad \left. - |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \right|^{p-2} \\ &\quad \left(\left(\sum_{v_k \in \mathcal{V}} \delta(v_k, e_q) w_I(v_k)^\alpha w_G(v_k)^\epsilon f(v_k) \right) \right. \\ &\quad \left. - |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \right) \\ &= \Delta_v^p f(v_i) \end{aligned}$$

Thus, the previously introduced definitions for the vertex gradient ∇_v , the vertex divergence div_v , and the vertex p -Laplacian Δ_v^p suffice the equality $\Delta_v^p f(v_i) = \text{div}_v(|\nabla_v f|^{p-2} \nabla_v f)(v_i)$ for all vertices $v_i \in \mathcal{V}$ and for all vertex functions $f \in \mathcal{H}(\mathcal{V})$. \square

3.5 Averaging Operators on Unoriented Hypergraphs

Instead of starting with a gradient definition in order to retrieve a feasible Laplacian operator for unoriented hypergraphs, we now want to define a Laplacian operator based on intuitive averaging. For this definition, a special vertex $v_{\tilde{q}}$ for every hyperedge $e_q \in \mathcal{E}_H$ is not necessary anymore. Before introducing the vertex averaging operator, we need the definition of the number of incident hyperedges.

Definition 16 [Number of incident hyperedges] For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$, the number of incident hyperedges of a given vertex $v_i \in \mathcal{V}$ is defined as:

$$\#\mathcal{E}_H(v_i) = |\{e_q \in \mathcal{E}_H \mid v_i \in e_q\}|. \quad (32)$$

Note: We call vertices $v_i \in \mathcal{V}$ with $\#\mathcal{E}_H(v_i) = 0$ isolated, since they are not connected to any other vertex $v_j \in \mathcal{V}$.

The averaging operator below aims at defining for a given vertex v_i the average value of a vertex function $f \in \mathcal{H}(\mathcal{V})$ by considering all hyperedges $e_q \in \mathcal{E}_H$, which v_i is a part of, and then averaging the vertex function over all vertices $v_j \in e_q$.

Definition 17 [Vertex averaging operator $\overline{\Delta}_v$] For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ without any isolated vertices $v_i \in \mathcal{V}$, i.e., $\#\mathcal{E}_H(v_i) > 0$ for all vertices $v_i \in \mathcal{V}$, we define the vertex averaging operator as:

$$\overline{\Delta}_v : \mathcal{H}(\mathcal{V}) \longrightarrow \mathcal{H}(\mathcal{V}) \quad f \longmapsto \overline{\Delta}_v f$$

$$\begin{aligned} \overline{\Delta}_v f : \mathcal{V} &\longrightarrow \mathbb{R} \quad v_i \longmapsto \overline{\Delta}_v f(v_i) = \\ &\frac{1}{\#\mathcal{E}_H(v_i)} \sum_{e_q \in \mathcal{E}_H} \delta(v_i, e_q) \\ &\frac{1}{|e_q|} \sum_{v_j \in \mathcal{V}} \delta(v_j, e_q) f(v_j). \end{aligned} \quad (33)$$

By using a simplified version of the inner product on the space of all vertex functions $\mathcal{H}(\mathcal{V})$ with $w_I \equiv 1$, we obtain an energy-conserving adjoint vertex averaging operator $\overline{\Delta}_v^*$.

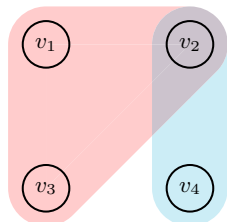
Definition 18 [Adjoint vertex averaging operator $\overline{\Delta}_v^*$] For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ without any isolated vertices and with the previously defined averaging operator $\overline{\Delta}_v$, the adjoint vertex averaging operator is given by:

$$\begin{aligned} \overline{\Delta}_v^* : \mathcal{H}(\mathcal{V}) &\longrightarrow \mathcal{H}(\mathcal{V}) \quad f \longmapsto \overline{\Delta}_v^* f \\ \overline{\Delta}_v^* f : \mathcal{V} &\longrightarrow \mathbb{R} \quad v_i \longmapsto \overline{\Delta}_v^* f(v_i) = \\ &\sum_{e_q \in \mathcal{E}_H} \delta(v_i, e_q) \frac{1}{|e_q|} \sum_{v_j \in \mathcal{V}} \\ &\delta(v_j, e_q) \frac{1}{\#\mathcal{E}_H(v_j)} f(v_j). \end{aligned} \quad (34)$$

Theorem 7 [Connection between vertex averaging operator $\overline{\Delta}_v$ and adjoint vertex averaging operator $\overline{\Delta}_v^*$] For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ without isolated vertices and with any two vertex functions $f, g \in \mathcal{H}(\mathcal{V})$, the vertex averaging operator $\overline{\Delta}_v$ and the adjoint vertex operator $\overline{\Delta}_v^*$ suffice the following equality

$$\begin{aligned} \langle g, \overline{\Delta}_v f \rangle_{\mathcal{H}(\mathcal{V})} &:= \sum_{v_i \in \mathcal{V}} g(v_i) \overline{\Delta}_v f(v_i) \\ &= \sum_{v_j \in \mathcal{V}} f(v_j) \overline{\Delta}_v^* g(v_j) \\ &=: \langle f, \overline{\Delta}_v^* g \rangle_{\mathcal{H}(\mathcal{V})}, \end{aligned} \quad (35)$$

where the inner product on the space of all vertex functions $\mathcal{H}(\mathcal{V})$ has the weight $w_I \equiv 1$.



$$\begin{aligned} \overline{\Delta}_v f(v_1) &= \frac{1}{1} \left(\frac{1}{3} (f(v_1) + f(v_2) + f(v_3)) \right) \\ \overline{\Delta}_v f(v_2) &= \frac{1}{2} \left(\frac{1}{3} (f(v_1) + f(v_2) + f(v_3)) + \frac{1}{2} (f(v_2) + f(v_4)) \right) \\ \overline{\Delta}_v f(v_3) &= \frac{1}{1} \left(\frac{1}{3} (f(v_1) + f(v_2) + f(v_3)) \right) \\ \overline{\Delta}_v f(v_4) &= \frac{1}{1} \left(\frac{1}{2} (f(v_2) + f(v_4)) \right) \end{aligned}$$

Proof Given an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ without any isolated vertices and two vertex functions $f, g \in \mathcal{H}(\mathcal{V})$, then the definitions of the vertex averaging operator $\overline{\Delta}_v$ and the adjoint vertex averaging operator $\overline{\Delta}_v^*$ yield the following:

$$\begin{aligned} \langle g, \overline{\Delta}_v f \rangle_{\mathcal{H}(\mathcal{V})} &= \sum_{v_i \in \mathcal{V}} g(v_i) \overline{\Delta}_v f(v_i) \\ &= \sum_{v_i \in \mathcal{V}} g(v_i) \frac{1}{\#\mathcal{E}_H(v_i)} \sum_{e_q \in \mathcal{E}_H} \delta(v_i, e_q) \frac{1}{|e_q|} \\ &\sum_{v_j \in \mathcal{V}} \delta(v_j, e_q) f(v_j) \\ &= \sum_{v_i \in \mathcal{V}} \sum_{e_q \in \mathcal{E}_H} \sum_{v_j \in \mathcal{V}} g(v_i) \frac{1}{\#\mathcal{E}_H(v_i)} \delta(v_i, e_q) \\ &\frac{1}{|e_q|} \delta(v_j, e_q) f(v_j) \\ &= \sum_{v_j \in \mathcal{V}} \sum_{e_q \in \mathcal{E}_H} \sum_{v_i \in \mathcal{V}} g(v_i) \frac{1}{\#\mathcal{E}_H(v_i)} \delta(v_i, e_q) \\ &\frac{1}{|e_q|} \delta(v_j, e_q) f(v_j) \\ &= \sum_{v_j \in \mathcal{V}} f(v_j) \sum_{e_q \in \mathcal{E}_H} \delta(v_j, e_q) \frac{1}{|e_q|} \\ &\sum_{v_i \in \mathcal{V}} \delta(v_i, e_q) \frac{1}{\#\mathcal{E}_H(v_i)} g(v_i) = \sum_{v_j \in \mathcal{V}} f(v_j) \overline{\Delta}_v^* g(v_j) \\ &= \langle f, \overline{\Delta}_v^* g \rangle_{\mathcal{H}(\mathcal{V})} \end{aligned}$$

Therefore, with the definitions of the vertex averaging operator $\overline{\Delta}_v$ and the adjoint vertex averaging operator $\overline{\Delta}_v^*$, the equality $\langle g, \overline{\Delta}_v f \rangle_{\mathcal{H}(\mathcal{V})} = \langle f, \overline{\Delta}_v^* g \rangle_{\mathcal{H}(\mathcal{V})}$ holds true for all vertex functions $f, g \in \mathcal{H}(\mathcal{V})$. \square

Example 3 [Vertex averaging operator does not conserve mean values] Given a set of vertices $\mathcal{V} = \{v_1, v_2, v_3, v_4\}$ and a set of hyperedges $\mathcal{E}_H = \{\{v_1, v_2, v_3\}, \{v_2, v_3, v_4\}\}$, then the vertex averaging operator $\overline{\Delta}_v$ on the unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ does not conserve the mean value for general vertex functions $f \in \mathcal{H}(\mathcal{V})$:

$$\begin{aligned}
\overline{\Delta}_v f(v_1) &= \frac{1}{1} \left(\frac{1}{3} (f(v_1) + f(v_2) + f(v_3)) \right) \\
\overline{\Delta}_v f(v_2) &= \frac{1}{2} \left(\frac{1}{3} (f(v_1) + f(v_2) + f(v_3)) \right. \\
&\quad \left. + \frac{1}{2} (f(v_2) + f(v_4)) \right) \\
\overline{\Delta}_v f(v_3) &= \frac{1}{1} \left(\frac{1}{3} (f(v_1) + f(v_2) + f(v_3)) \right) \\
\overline{\Delta}_v f(v_4) &= \frac{1}{1} \left(\frac{1}{2} (f(v_2) + f(v_4)) \right) \\
\overline{\Delta}_v f(v_1) + \overline{\Delta}_v f(v_2) + \overline{\Delta}_v f(v_3) + \overline{\Delta}_v f(v_4) &+ \\
&= \frac{5}{6} f(v_1) + \frac{19}{12} f(v_2) + \frac{5}{6} f(v_3) + \frac{3}{4} f(v_4) \\
&\neq f(v_1) + f(v_2) + f(v_3) + f(v_4)
\end{aligned}$$

In contrast to this, the adjoint vertex averaging operator conserves the overall energy for all vertex functions $f \in \mathcal{H}(\mathcal{V})$.

Theorem 8 [Adjoint vertex averaging operator $\overline{\Delta}_v^*$ conserves mean values] For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ without isolated vertices and with any vertex function $f \in \mathcal{H}(\mathcal{V})$, the adjoint vertex averaging operator $\overline{\Delta}_v^*$ conserves the mean value of f , and hence, the following equality holds:

$$\sum_{v_i \in \mathcal{V}} \overline{\Delta}_v^* f(v_i) = \sum_{v_i \in \mathcal{V}} f(v_i). \quad (36)$$

Proof Given an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ without any isolated vertices and a vertex function $f \in \mathcal{H}(\mathcal{V})$, then the following reformulations hold true:

$$\begin{aligned}
&\sum_{v_i \in \mathcal{V}} \overline{\Delta}_v^* f(v_i) \\
&= \sum_{v_i \in \mathcal{V}} \sum_{e_q \in \mathcal{E}_H} \delta(v_i, e_q) \frac{1}{|e_q|} \\
&\sum_{v_j \in \mathcal{V}} \delta(v_j, e_q) \frac{1}{\# \mathcal{E}_H(v_j)} f(v_j) \\
&= \sum_{v_i \in \mathcal{V}} \sum_{e_q \in \mathcal{E}_H} \sum_{v_j \in \mathcal{V}} \delta(v_i, e_q) \\
&\quad \frac{1}{|e_q|} \delta(v_j, e_q) \frac{1}{\# \mathcal{E}_H(v_j)} f(v_j) \\
&= \sum_{v_j \in \mathcal{V}} \sum_{e_q \in \mathcal{E}_H} \sum_{v_i \in \mathcal{V}} \delta(v_i, e_q) \\
&\quad \frac{1}{|e_q|} \delta(v_j, e_q) \frac{1}{\# \mathcal{E}_H(v_j)} f(v_j) \\
&= \sum_{v_j \in \mathcal{V}} f(v_j) \frac{1}{\# \mathcal{E}_H(v_j)} \sum_{e_q \in \mathcal{E}_H} \delta(v_j, e_q)
\end{aligned}$$

$$\begin{aligned}
&\frac{1}{|e_q|} \sum_{v_i \in \mathcal{V}} \delta(v_i, e_q) \\
&= \sum_{v_j \in \mathcal{V}} f(v_j) \frac{1}{\# \mathcal{E}_H(v_j)} \sum_{e_q \in \mathcal{E}_H} \delta(v_j, e_q) \\
&= \sum_{v_j \in \mathcal{V}} f(v_j) \frac{1}{\# \mathcal{E}_H(v_j)} \# \mathcal{E}_H(v_j) \\
&= \sum_{v_j \in \mathcal{V}} f(v_j)
\end{aligned}$$

Hence, the presented adjoint vertex averaging operator $\overline{\Delta}_v^*$ conserves the overall energy on any given unoriented hypergraph UH for all vertex functions $f \in \mathcal{H}(\mathcal{V})$. \square

Let us put together the knowledge gained from the above analysis. Since we used the simple Euclidean scalar product and showed that the adjoint operator conserves the mean value, we can look at $\overline{\Delta}_v - I$ as a suitable operator for a scale space analysis, somehow introducing a non-self-adjoint version of the Laplacian. The energy conservation of the adjoint shows that indeed $\overline{\Delta}_v$ has eigenvalue one with constant eigenfunction. Thus, the evolution equation with operator $\overline{\Delta}_v - I$ is expected to converge to a constant state and yield a suitable scale space, which we will investigate further below.

Indeed, the averaging operator is self-adjoint if $\# \mathcal{E}_H(v_i)$ is constant on the set of vertices \mathcal{V} , and in this case $\overline{\Delta}_v - I$ has the structure of a normal graph Laplacian.

Lemma 1 Given an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ with $\# \mathcal{E}_H(v_i) = \# \mathcal{E}_H(v_j)$ for all vertices $v_i, v_j \in \mathcal{V}$. Then for all vertex functions $f \in \mathcal{H}(\mathcal{V})$, the operator $\overline{\Delta}_v f - f$ is equivalent to the graph Laplacian on a weighted oriented graph for an arc (v_i, v_j) from vertex v_i to vertex v_j with the particular weight function

$$\begin{aligned}
w(v_i, v_j) &:= \\
&\frac{1}{\# \mathcal{E}_H(v_i)} \sum_{e_q \in \mathcal{E}_H} \frac{1}{|e_q|} \delta(v_i, e_q) \delta(v_j, e_q).
\end{aligned}$$

Proof For an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ with $\# \mathcal{E}_H(v_i) = \# \mathcal{E}_H(v_j)$ for all vertices $v_i, v_j \in \mathcal{V}$, the equivalence becomes apparent from a simple change of summation:

$$\begin{aligned}
&(\overline{\Delta}_v f - f)(v_i) \\
&= \left(\frac{1}{\# \mathcal{E}_H(v_i)} \sum_{e_q \in \mathcal{E}_H} \delta(v_i, e_q) \frac{1}{|e_q|} \sum_{v_j \in \mathcal{V}} \delta(v_j, e_q) f(v_j) \right) \\
&\quad - f(v_i)
\end{aligned}$$

$$\begin{aligned}
&= \left(\frac{1}{\#\mathcal{E}_H(v_i)} \sum_{e_q \in \mathcal{E}_H} \delta(v_i, e_q) \frac{1}{|e_q|} \sum_{v_j \in \mathcal{V}} \delta(v_j, e_q) f(v_j) \right) \\
&\quad - \frac{1}{\#\mathcal{E}_H(v_i)} \sum_{e_q \in \mathcal{E}_H} \delta(v_i, e_q) \frac{1}{|e_q|} f(v_i) \\
&= \left(\frac{1}{\#\mathcal{E}_H(v_i)} \sum_{e_q \in \mathcal{E}_H} \delta(v_i, e_q) \frac{1}{|e_q|} \sum_{v_j \in \mathcal{V}} \delta(v_j, e_q) f(v_j) \right) \\
&\quad - \left(\frac{1}{\#\mathcal{E}_H(v_i)} \sum_{e_q \in \mathcal{E}_H} \delta(v_i, e_q) \frac{1}{|e_q|} \sum_{v_j \in \mathcal{V}} \delta(v_j, e_q) \right) f(v_i) \\
&= \frac{1}{\#\mathcal{E}_H(v_i)} \sum_{e_q \in \mathcal{E}_H} \delta(v_i, e_q) \frac{1}{|e_q|} \\
&\quad \sum_{v_j \in \mathcal{V}} \delta(v_j, e_q) (f(v_j) - f(v_i)) \\
&= \sum_{e_q \in \mathcal{E}_H} \sum_{v_j \in \mathcal{V}} \left(\frac{1}{\#\mathcal{E}_H(v_i)} \delta(v_i, e_q) \frac{1}{|e_q|} \right. \\
&\quad \left. \delta(v_j, e_q) (f(v_j) - f(v_i)) \right) \\
&= \sum_{v_j \in \mathcal{V}} \sum_{e_q \in \mathcal{E}_H} \left(\frac{1}{\#\mathcal{E}_H(v_i)} \delta(v_i, e_q) \frac{1}{|e_q|} \right. \\
&\quad \left. \delta(v_j, e_q) (f(v_j) - f(v_i)) \right) \\
&= \sum_{v_j \in \mathcal{V}} \left(\frac{1}{\#\mathcal{E}_H(v_i)} \sum_{e_q \in \mathcal{E}_H} \frac{1}{|e_q|} \right. \\
&\quad \left. \delta(v_i, e_q) \delta(v_j, e_q) \right) (f(v_j) - f(v_i)) \\
&= \sum_{v_j \in \mathcal{V}} w(v_i, v_j) (f(v_j) - f(v_i))
\end{aligned}$$

The term in the last row is exactly the traditional graph Laplace operator of vertex v_i for an oriented normal graph with a vertex function f (see [7] Remark 7.12 (Parameter choice for the vertex p -Laplacian operator)), where for the arc weight w it holds true that $w(v_i, v_j) = 0$ if the arc (v_i, v_j) does not exist in the oriented normal graph. \square

Let us mention that the weighted oriented normal graph we obtain above could be considered the easiest map from an unoriented hypergraph to a weighted graph, since the weights essentially count the number of hyperedges $e_q \in \mathcal{H}(\mathcal{V})$ two vertices $v_i, v_j \in \mathcal{V}$ have in common.

4 Scale Spaces Based on Hypergraph p -Laplacians

In the following, we discuss PDEs based on the family of p -Laplace and averaging operators on hypergraphs intro-

duced in Sect. 3, which can be used for modeling information flow in social networks with oriented hypergraphs as well as performing image processing based on both oriented and unoriented hypergraphs.

4.1 Modeling Information Flow Using Oriented Hypergraphs

For analyzing information flow on social networks with oriented hypergraphs, we consider two different PDE systems modeling diffusion processes. We start with investigating the scale space for the p -Laplacian operator, i.e., the gradient flow of the p -Laplacian energy:

$$\begin{aligned}
\frac{\partial f}{\partial t}(v_i, t) &= \Delta_v^p f(v_i, t), \quad v_i \in \mathcal{V}, t \in (0, \infty) \\
f(v_i, 0) &= f_0(v_i), \quad v_i \in \mathcal{V}.
\end{aligned} \tag{37}$$

Solving (37) for every time step $t \in (0, \infty)$ amounts to computing the information flow between vertices of the oriented hypergraph along the respective hyperarcs.

Note that although there are no explicit boundaries in oriented hypergraphs, we can interpret the above problem as the homogeneous Neumann boundary problem. Due to the properties of the proposed family of hypergraph p -Laplace operators, it is easy to see that the mean-value of f is conserved in time and we can naturally interpret the evolution as a scale space toward coarser and coarser scales on the graph. Moreover, the general asymptotic of gradient flows for p -homogeneous energies (cf. [2]) yields that $f \rightarrow \bar{f}$ as $t \rightarrow \infty$, with \bar{f} being the mean value of f_0 . Moreover, the rescaled quantity $g = \frac{f - \bar{f}}{\|f - \bar{f}\|}$ converges to a multiple of a second eigenfunction for generic initial values.

Similar to the Neumann boundary problem, we can also introduce a Dirichlet-type problem, where the Dirichlet boundary $\partial\mathcal{V} \subset \mathcal{V}$ denotes a subset of the vertex set \mathcal{V} of the oriented hypergraph, for which we introduce boundary values and keep them fixed over time. The corresponding stationary solution is not necessarily constant

$$\begin{aligned}
\Delta_v^p f(v_i) &= 0, \quad v_i \in \mathring{\mathcal{V}}, \\
f(v_j) &= F_j, \quad v_j \in \partial\mathcal{V}.
\end{aligned} \tag{38}$$

Then, we aim at solving the p -Laplace equation on the complementary vertex set $\mathring{\mathcal{V}} := \mathcal{V} \setminus \partial\mathcal{V}$ of the oriented hypergraph. Instead of solving (38) directly, we solve the hyperbolic PDE model (37) on the vertex set $\mathring{\mathcal{V}}$, while keeping the vertex function $f \in \mathcal{H}(\mathcal{V})$ fixed on the boundary set $\partial\mathcal{V}$. The reason for this approach is that any stationary solution of (37) on $\mathring{\mathcal{V}}$ with fixed boundary values is also a solution to the p -Laplace equation in (38). To solve the two proposed PDE models discussed above, we numerically have to solve the initial value problem in (37). For this sake, we

employ a forward-Euler time discretization with fixed time step size $\tau > 0$ and use the renormalized variable g to observe convergence to a nontrivial eigenfunction. This leads to the following explicit iteration scheme:

$$f_{n+1}(v_i) = f_n(v_i) + \tau \cdot \Delta_v^p f_n(v_i). \quad (39)$$

4.2 Image Processing Using Unoriented Hypergraphs

To perform image processing for grayscale images defined on regular grids, we consider a PDE system that can be interpreted as an initial value problem for the vertex averaging operator introduced in Definition 17. In particular, we are interested in solving the following initial value problem

$$\begin{aligned} & \frac{\partial f}{\partial t}(v_i, t) - \lambda \cdot (f_0(v_i) - f(v_i, t)) \\ &= \overline{\Delta_v} f(v_i, t) - f(v_i, t), \quad v_i \in \mathcal{V}, t \in (0, \infty) \\ & f(v_i, 0) = f_0(v_i), \quad v_i \in \mathcal{V}. \end{aligned} \quad (40)$$

Note that in contrast to the initial value problem modeling opinion formation in social networks in (37), here we introduce an additional data fidelity term that penalizes strong deviations from the noisy image, represented by the initial vertex function $f_0 \in \mathcal{H}(\mathcal{V})$. The influence of this data fidelity term can be controlled by a fixed parameter $\lambda > 0$ that allows to realize a trade-off between smoothing the perturbed image pixels via the hypergraph vertex averaging operator and staying close to the initial image. In classical variational regularization, this would correspond to the gradient flow of the least-squares fidelity augmented with a regularization energy scaled with regularization parameter $\frac{1}{\lambda}$. However, since the averaging operator is not self-adjoint, the corresponding term in (40) cannot arise in the gradient flow of an associated energy functional. Nonetheless, the diffusive nature of $\overline{\Delta_v} f - f$ induces an interpretation as regularization albeit in nonvariational setting, similar, e.g., to the inpainting model in [3].

Once again we use a forward-Euler time discretization with fixed time step size $\tau > 0$, which is chosen small enough to fulfill the CFL stability conditions. Following this approach, we derive the following iterative scheme for image processing using the hypergraph vertex averaging operator:

$$\begin{aligned} f_{n+1}(v_i) &= f_n(v_i) + \tau \cdot (\overline{\Delta_v} f_n(v_i) - f_n(v_i) \\ &\quad + \lambda \cdot (f_0(v_i) - f_n(v_i))). \end{aligned} \quad (41)$$

5 Numerical Experiments

In this section, we present the results of our numerical experiments when using the hypergraph operators introduced in Sect. 3 for two different applications. In particular, we first

discuss how the oriented hypergraph p -Laplacian operator can be used to model opinion formation in social networks. Furthermore, we apply the vertex averaging operator of unoriented hypergraphs for the task of image processing, and we provide results that can be used both for image denoising and for segmentation tasks.

5.1 Opinion Formation in Social Networks

In the following, we present the results of our numerical experiments in which we solve the two PDEs (37) and (38) by using the explicit forward-Euler discretization scheme until the relative change between two iterations is smaller than $\epsilon := 10^{-6}$. We choose τ in (39) small enough to fulfill the CFL condition for numerical stability. This leads to very small time steps for the iteration scheme in the cases $1 \leq p < 2$.

For our numerical experiments, we use the Twitter data set provided by Stanford University [10]. It consists of 41, 652, 230 vertices (users) and 1, 468, 364, 884 arcs (oriented pairwise connections indicating that one person follows another). Due to the size of the data set, we restrict our numerical experiments to a comparatively small sub-network within the first 1, 000, 000 lines of the Twitter input data. We chose a sub-network of individuals such that all users are directly or indirectly linked to each other to avoid cliques of individuals, which are not connected to the rest of the sub-network and thus also not influenced by users outside their small circle. Moreover, we ensure that each sub-network includes an opinion leader with a large number of followers in the sub-network. Therefore, we can observe how one influential user impacts the opinion of the rest. In order to generate hyperarcs from the given arcs, we put one Twitter user as a singleton output vertex set and summarize all followers of this user as the set of input vertices. This especially allows highlighting the effect of opinion leaders, for instance famous people with a large group of followers on Twitter.

We simulate the opinion of all individuals in the social network toward an imaginary hypothesis by a vertex function $f: \mathcal{V} \times [0, \infty) \rightarrow [-1, 1]$, which can be interpreted as the following. If an individual believes the hypothesis the corresponding value of the vertex function is positive (with 1 being the strongest level of trust), while for an individual that opposes the hypothesis the corresponding value of the vertex function is negative (with -1 being the strongest level of distrust).

For the **boundary value problem** (38), we initialize the opinion of all individuals in a social network by setting the vertex function f to zero, which can be interpreted as having no opinion toward an imaginary hypothesis. We now simulate information flow in the social network by giving two opinion leaders (i.e., vertices with many followers) two opposing opinions toward this hypothesis and setting the respective

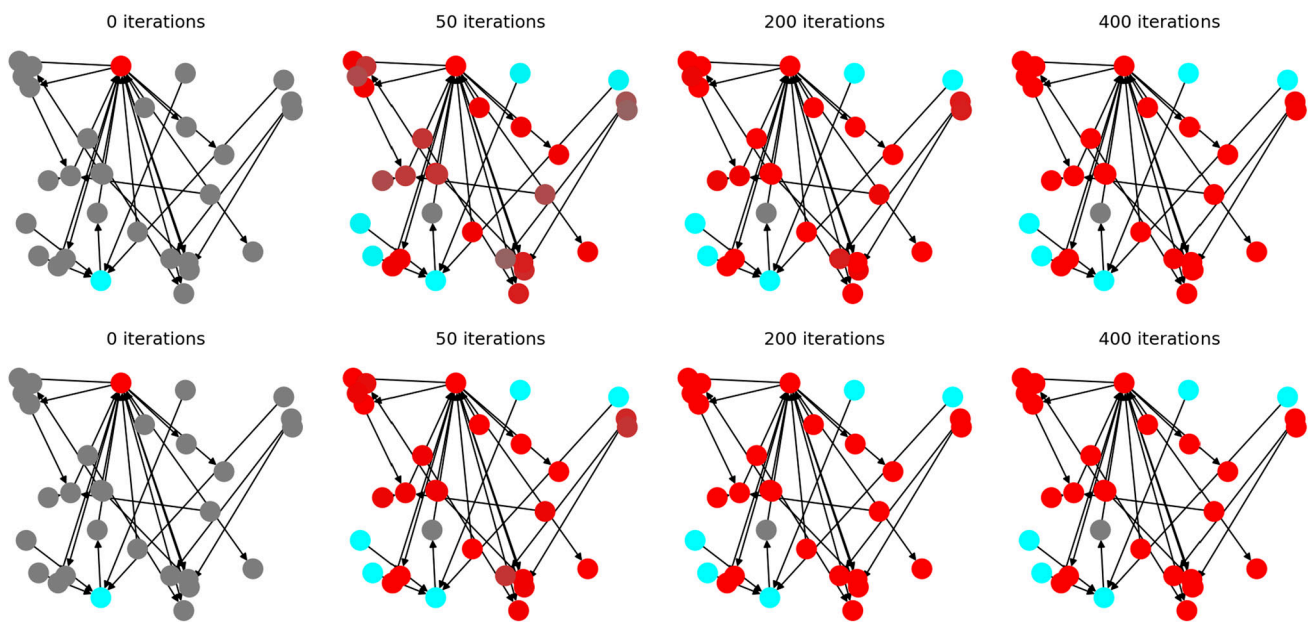


Fig. 1 Solution of the boundary value problem of graph (top) and hypergraph (bottom) p -Laplace operator for $p = 2$

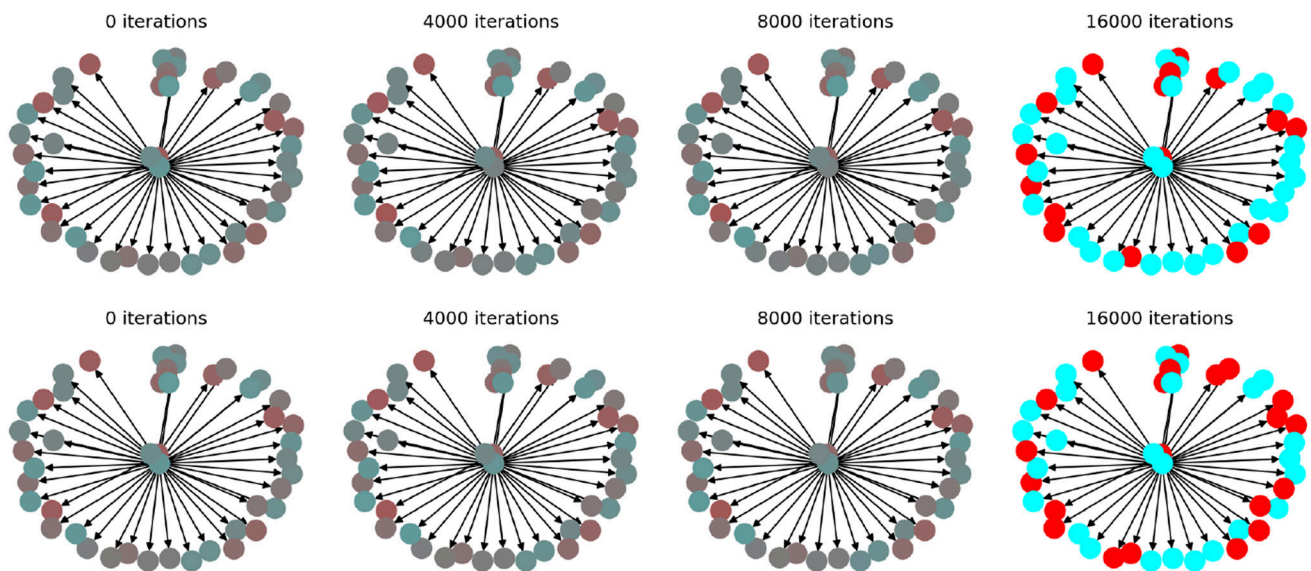


Fig. 2 Second eigenfunction of graph (top) and hypergraph (bottom) p -Laplace operator for $p = 1$ with thresholding at 0 after 16,000 iterations

values of the vertex function to -1 and 1 . We keep these values fixed as a form of Dirichlet boundary conditions. By using the explicit forward-Euler discretization scheme to solve the boundary value problem for $p = 2$, the opinion of the two dedicated individuals is propagated in the social network as can be seen in Fig. 1. We initialize the vertex function equally for the oriented normal graph (top row) and the oriented hypergraph (bottom row) and calculate the diffusion process until convergence. As can be seen, in both cases the opinion is propagated in the social network based on the underlying network topology and the final state is equivalent

for both the normal graph and the hypergraph experiment. However, as can be observed, information within the hypergraph is distributed at a higher rate compared to the normal graph and thus converging faster. This is due to the fact that opinion leaders in a normal graph have a less direct impact on their followers compared to the hypergraph case, where the follower's belief $f(v_i)$ is scaled with $\frac{1}{|a_q^{in}|}$, where $|a_q^{in}|$ is the number of followers of the individual user. This can be seen in (19) since in our modeling for this application the parameter a_q^{out} is set to 1.

For the **initial value problem** (37), we choose $p = 1$ and a sufficiently small time step size $\tau > 0$ to guarantee stability of the corresponding iteration scheme (39). We initialize each individual's opinion $f_0(v_i)$ randomly with a uniform distribution in the interval $[-1, 1]$. Additionally, we make sure that the vertex function f is initialized with average 0 and normalized. As can be observed in Fig. 2, the information flow in the social network converges to a second eigenfunction of both the graph p -Laplacian (top row) and the hypergraph p -Laplacian (bottom row). For both cases, we thresholded at 0 after 16, 000 iterations to induce a spectral clustering of the opposing opinions and hence separating the social network into smaller communities based on the topology of the network (i.e., the relationship of following an individual). The resulting second eigenfunctions differ significantly with respect to the underlying topology of the oriented normal graph and the oriented hypergraph. This yields potential for further analysis and experiments in other applications, e.g., segmentation of images via spectral clustering.

5.2 Local and Nonlocal Image Processing

In the following, we discuss how the proposed hypergraph differential operators can be applied to image processing tasks. By modeling pixels of an image with the help of normal graphs or hypergraphs instead of a regular grid, it is possible to not only represent local relationships of adjacent pixels, but also nonlocal relationships based on the image's content. For example, one could link image pixels that are relatively far from each other in the image, but share a similar image texture in their respective neighborhood.

Given an image $\tilde{I} \in \mathbb{R}^{n \times m}$ of height $n \in \mathbb{N}$ and width $m \in \mathbb{N}$ perturbed by a normal distributed noise signal $v \sim \mathcal{N}(0, \sigma^2)$, a typical task in image processing is to recover a noise-free image $I \in \mathbb{R}^{n \times m}$ from

$$\tilde{I} = I + v. \quad (42)$$

This task can be interpreted as an inverse problem known as *image denoising*. In our numerical experiments we restrict ourselves to grayscale images for the sake of simplicity.

To perform image denoising, we first model the relationship between the image pixels with an unoriented hypergraph. In particular, we represent each image pixel as a vertex $v_i \in \mathcal{V}$ of the oriented hypergraph and interpret the pixel grayscale intensities as the values of the vertex function $f \in \mathcal{H}(\mathcal{V})$ with $f: \mathcal{V} \rightarrow [0, 255]$. Here, 0 represents the lowest signal intensity (i.e., black pixels) and 255 represents the highest signal intensity (i.e., white pixels). We construct the hyperedges of the hypergraph as described in detail below.

For our numerical experiments, we chose a grayscale image I of size 225×400 pixels of a flower field that contains

image features at different scales. We added random Gaussian noise v with mean $\mu = 0$ and variance $\sigma^2 = 150$ to every image pixel to generate an artificially perturbed image \tilde{I} . Both the unperturbed image I and the noisy variant \tilde{I} are illustrated in Fig. 3

We construct an initial vertex function $f_0 \in \mathcal{H}(\mathcal{V})$ from the noisy image \tilde{I} .

We performed two different experiments for image denoising using the introduced unoriented hypergraph vertex average operator $\overline{\Delta}_v$ from definition 17. In the first experiment, we perform **local image processing** by constructing a hyperedge of the unoriented hypergraph from the vertices that model the direct four pixel neighbors of any image pixel. This corresponds to traditional image processing methods on regular grids as performed, e.g., in [3]. For boundary pixels of the image, we use an analogue of Neumann zero boundary conditions, i.e., we assume the image is extended constantly. This results in a total of $225 \cdot 400 = 90000$ hyperedges for the unoriented hypergraph, where each hyperedge $e_q \in \mathcal{E}_H$ can be directly associated with the corresponding image pixel.

In the first case, we compare the results of using the iterative scheme (41) with and without data fidelity term, for which we each performed 100 iterations in our numerical experiments. To investigate the influence of the data fidelity term, we first fixed the time step size as $\tau := 0.1$ and varied the regularization parameter $\lambda > 0$. The left column of Fig. 4 shows that with decreasing value of $\lambda > 0$, the smoothing effect of the vertex averaging operator increases, leading to less noisy images. On the other hand, the edges of image features get more and more blurry as can be expected in this case. In another setting, we remove the data fidelity term by setting $\lambda := 0$ and hence we investigate the corresponding evolution equation of the vertex averaging operator. Here, we varied the time step size parameter $\tau > 0$ to compare different results for a fixed number of iterations. As can be seen in the right column of Fig. 4, we recover the scale spaces of the operator $\overline{\Delta}_v$. With increasing time step size $\tau > 0$, we observe more and more coarse image features induced by the local averaging effect of the operator.

In the second numerical experiment, we perform **nonlocal image processing** by constructing the hyperedges of the unoriented hypergraph not from the local neighborhood of an image pixel, but by regarding the pixel intensities of the image. By this we are able to assemble vertices in hyperedges that correspond to image pixels which can be anywhere in the image and hence we gain a nonlocal vertex averaging operator. In particular, we construct for each vertex $v_i \in \mathcal{V}$ a hyperedge containing all vertices for which the value of the vertex function f is similar. For this, we choose a threshold $\epsilon > 0$ and add all vertices $v_j \in \mathcal{V}$ to the hyperedge induced by the vertex $v_i \in \mathcal{V}$ for which the distance is small enough,

Fig. 3 Illustration of the original grayscale image I (left) and the artificially perturbed image \tilde{I} (right) used for image denoising



Fig. 4 Solution of the initial value problem (40) for the average hypergraph operator on a **local hypergraph** with data fidelity term (left) and without data fidelity term (right)



(a) $\tau = 0.1, \lambda = 10$, 100 iterations

(b) $\tau = 0.01, \lambda = 0$, 100 iterations



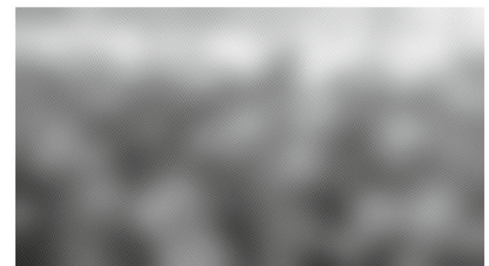
(c) $\tau = 0.1, \lambda = 1$, 100 iterations

(d) $\tau = 0.25, \lambda = 0$, 100 iterations



(e) $\tau = 0.1, \lambda = 0.1$, 100 iterations

(f) $\tau = 1, \lambda = 0$, 100 iterations



(g) $\tau = 0.1, \lambda = 0.01$, 100 iterations

(h) $\tau = 2, \lambda = 0$, 100 iterations

i.e.,

$$|f(v_i) - f(v_j)| < \epsilon. \quad (43)$$

We chose $\epsilon := 40$ for the experiments presented in this setting. With this approach, we nonlocally group image pixels with similar grayscale intensities. Since pixels with equal grayscale intensity lead to similar or even equal hyperedges, we treat hyperedges uniquely, i.e., without any duplicates in $\mathcal{H}(\mathcal{E})$.

As in the case of local image processing described above, we compared the results of using the iterative scheme (41) with and without data fidelity term, for which we performed 100 iterations in our numerical experiments. We started our experiments by fixing the time step size as $\tau := 1$ and varying the regularization parameter $\lambda > 0$ in order to investigate the influence of the data fidelity term. The left column in Fig. 5 shows that with decreasing value of $\lambda > 0$, the results deviate more and more from the initial image. Furthermore, one can observe that the amount of different pixel intensities decreases more and more until the resulting image shows only one grayscale intensity in the last row. At the same time, edges of image features stay sharp as there is no averaging operation across the boundaries of image regions with strongly varying grayscale intensities. Secondly, we again removed the influence of the data fidelity term entirely by setting $\lambda := 0$. By varying the time step size parameter $\tau > 0$, we can compare different scales of the resulting scale space for a fixed number of iterations of the nonlocal hypergraph vertex averaging operator Δ_v . As in the experiment with data fidelity, we observe that the amount of different pixel intensities decreases rapidly with increasing time step size $\tau > 0$, leading to a grouping of image pixels into image regions with similar grayscale intensities, until eventually all image pixels have the same grayscale value in the last row. This expected behavior can be leveraged for other image processing tasks in which grouping of image pixels is needed, e.g., in *image compression* or *segmentation*.

6 Conclusions

In this paper, we derived various variants of differential operators and a family of p -Laplacian operators on hypergraphs, which generalize known graph and hypergraph operators from literature. In particular, we considered gradient, adjoint and p -Laplacian definitions both in the case of oriented and unoriented hypergraphs.

The resulting operators on oriented hypergraphs and the associated scale space flows can be employed for modeling information flows or performing spectral analysis in social networks, where we can directly incorporate group relationships via hyperarcs. Moreover, the proposed averaging

operators and p -Laplacians for unoriented hypergraphs enable performing local and nonlocal image processing with results that can be used for segmentation tasks. Preliminary results indicate a great potential for future research. Interesting further questions, in addition to a more detailed study of spectral clustering, are, e.g., the relation between hypergraph gradients and higher-order methods for partial differential equations or the definition of distance functions on hypergraphs via eigenfunctions of the infinity-Laplacian.

Another promising direction is to investigate the influence of non-constant weight functions of the hypergraphs used in our numerical experiments. In particular, we propose to define the weights of hyperedges based on the variance of the vertex function values for the vertices included in the respective hyperedge. This should further improve the results of both local and nonlocal image processing.

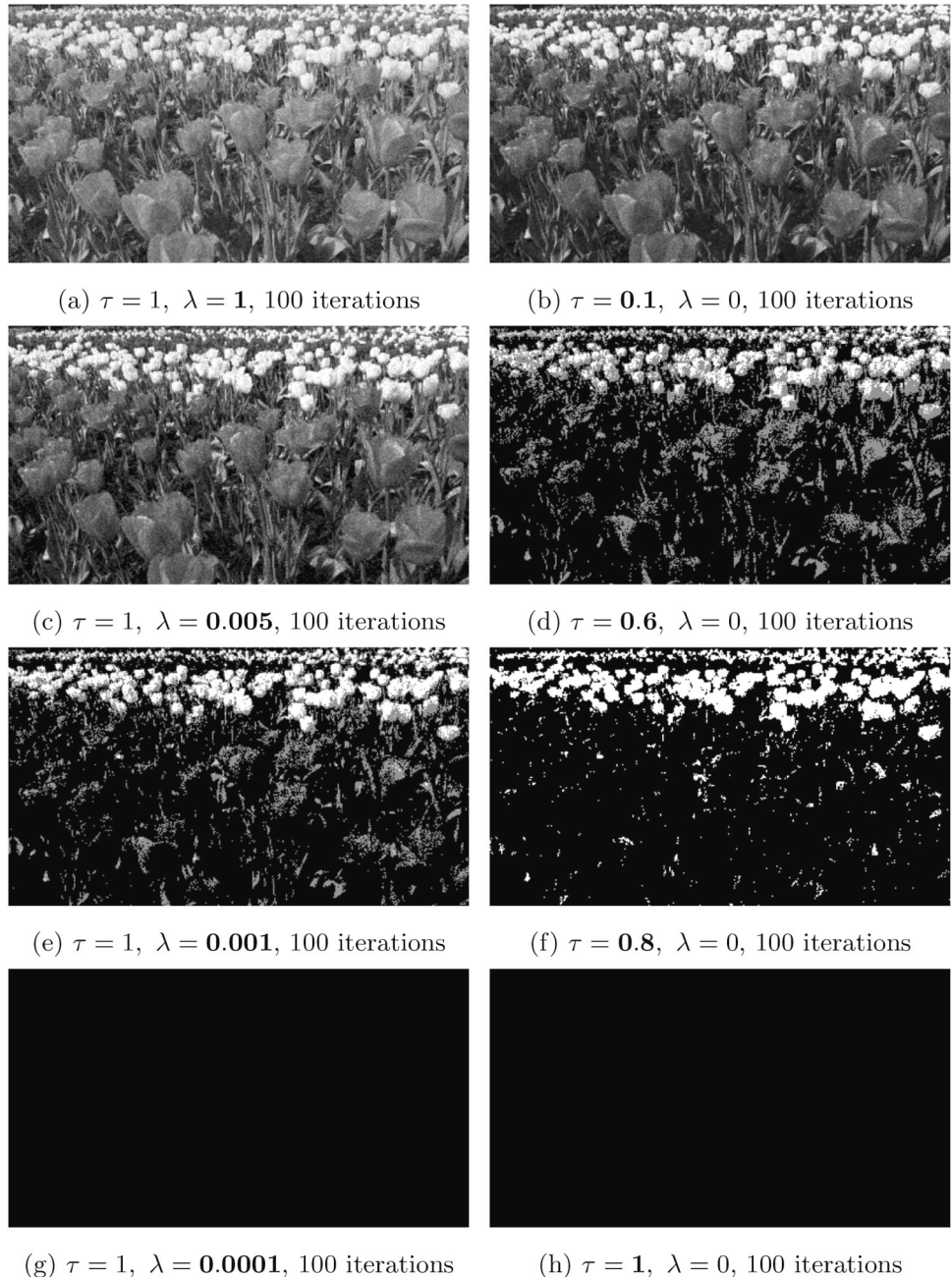
Due to the overarching success of learning-based methods in many areas of image processing and data analysis, it seems obvious that further developments in this directions are in place for hypergraph structures. We believe that our work can provide a foundation for hypergraph neural networks generalizing the recently celebrated graph neural network structures with unforeseen opportunities.

A Proof of Theorem 5

Proof Given an unoriented hypergraph $UH = (\mathcal{V}, \mathcal{E}_H)$ with vertex weight functions w_I and w_G , and hyperedge weight functions W_I and W_G , a vertex function $f \in \mathcal{H}(\mathcal{V})$, and a hyperedge function $F \in \mathcal{H}(\mathcal{E}_H)$ then using the definitions of the inner product on $\mathcal{H}(\mathcal{E}_H)$ and of the vertex gradient ∇_v results in:

$$\begin{aligned} \langle G, \nabla_v f \rangle_{\mathcal{H}(\mathcal{E}_H)} &= \sum_{e_q \in \mathcal{E}_H} W_I(e_q)^\beta G(e_q) \nabla_v f(e_q) \\ &= \sum_{e_q \in \mathcal{E}_H} W_I(e_q)^\beta G(e_q) W_G(e_q)^\gamma \\ &\quad \left(\left(\sum_{v_i \in \mathcal{V}} \delta(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) \right) \right. \\ &\quad \left. - |e_q| w_I(v_{\bar{q}})^\alpha w_G(v_{\bar{q}})^\eta f(v_{\bar{q}}) \right) \\ &= \sum_{e_q \in \mathcal{E}_H} W_I(e_q)^\beta G(e_q) W_G(e_q)^\gamma \\ &\quad \left(\sum_{v_i \in \mathcal{V}} \delta(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) \right) \\ &\quad - W_I(e_q)^\beta G(e_q) W_G(e_q)^\gamma \end{aligned}$$

Fig. 5 Solution of the initial value problem (40) for the average hypergraph operator on a **nonlocal hypergraph** with data fidelity term (left) and without data fidelity term (right)



$$\begin{aligned}
 & |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \\
 &= \sum_{e_q \in \mathcal{E}_H} \sum_{v_i \in \mathcal{V}} W_I(e_q)^\beta G(e_q) W_G(e_q)^\gamma \\
 &\quad \delta(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) \\
 &\quad - \sum_{e_r \in \mathcal{E}_H} W_I(e_r)^\beta G(e_r) W_G(e_r)^\gamma \\
 &\quad |e_r| w_I(v_{\tilde{r}})^\alpha w_G(v_{\tilde{r}})^\eta f(v_{\tilde{r}}) \\
 &= \sum_{e_q \in \mathcal{E}_H} \sum_{v_i \in \mathcal{V}} W_I(e_q)^\beta G(e_q) W_G(e_q)^\gamma
 \end{aligned}$$

$$\begin{aligned}
 & \delta(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) \\
 & - \sum_{e_r \in \mathcal{E}_H} \sum_{v_j \in \mathcal{V}} \tilde{\delta}(v_j, e_r) W_I(e_r)^\beta G(e_r) W_G(e_r)^\gamma \\
 & |e_r| w_I(v_{\tilde{r}})^\alpha w_G(v_{\tilde{r}})^\eta f(v_{\tilde{r}})
 \end{aligned}$$

The last reformulation holds true, because for each hyper-edge $e_r \in \mathcal{E}_H$ there exists exactly one special vertex $v_j \in \mathcal{V}$ with $v_j = v_{\tilde{q}}$ and thus the additional sum in combination with the characteristic function has no effect.

$$\begin{aligned}
&= \sum_{e_q \in \mathcal{E}_H} \sum_{v_i \in \mathcal{V}} \left(W_I(e_q)^\beta G(e_q) W_G(e_q)^\gamma \right. \\
&\quad \delta(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) \\
&\quad \left. - \tilde{\delta}(v_i, e_q) W_I(e_q)^\beta G(e_q) W_G(e_q)^\gamma \right. \\
&\quad \left. |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \right) \\
&= \sum_{v_i \in \mathcal{V}} \sum_{e_q \in \mathcal{E}_H} \left(W_I(e_q)^\beta G(e_q) W_G(e_q)^\gamma \right. \\
&\quad \delta(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) \\
&\quad \left. - \tilde{\delta}(v_i, e_q) W_I(e_q)^\beta G(e_q) W_G(e_q)^\gamma \right. \\
&\quad \left. |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \right) \\
&= \sum_{v_i \in \mathcal{V}} \sum_{e_q \in \mathcal{E}_H} W_I(e_q)^\beta W_G(e_q)^\gamma G(e_q) \\
&\quad \left(\delta(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) \right. \\
&\quad \left. - \tilde{\delta}(v_i, e_q) |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) \right) \\
&= \sum_{v_i \in \mathcal{V}} \sum_{e_q \in \mathcal{E}_H} W_I(e_q)^\beta W_G(e_q)^\gamma G(e_q) \\
&\quad \left(\delta(v_i, e_q) w_G(v_i)^\epsilon - \tilde{\delta}(v_i, e_q) |e_q| w_G(v_i)^\eta \right) \\
&\quad w_I(v_i)^\alpha f(v_i)
\end{aligned}$$

The last equality is based on the following argument: For a vertex $v_i \in \mathcal{V}$ and a hyperedge $e_q \in \mathcal{E}_H$, there are four cases:

- (1) $\delta(v_i, e_q) = \tilde{\delta}(v_i, e_q) = 0 \implies$
 $\delta(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) - \tilde{\delta}(v_i, e_q) |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) = 0$
- (2) $\delta(v_i, e_q) = \tilde{\delta}(v_i, e_q) = 1 \implies v_i = v_{\tilde{q}}$ and
 $\delta(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) - \tilde{\delta}(v_i, e_q) |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) =$
 $w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) - |e_q| w_I(v_i)^\alpha w_G(v_i)^\eta f(v_i) =$
 $(w_G(v_i)^\epsilon - |e_q| w_G(v_i)^\eta) w_I(v_i)^\alpha f(v_i)$
- (3) $\delta(v_i, e_q) = 1$ and $\tilde{\delta}(v_i, e_q) = 0 \implies$
 $\delta(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) - \tilde{\delta}(v_i, e_q) |e_q| w_I(v_{\tilde{q}})^\alpha w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) =$
 $\delta(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i)$
- (4) $\delta(v_i, e_q) = 0$ and $\tilde{\delta}(v_i, e_q) = 1$ impossible by definition

Hence the following equality holds true in all four cases, which verifies the feasibility of the reformulation above:

$$\begin{aligned}
&\delta(v_i, e_q) w_I(v_i)^\alpha w_G(v_i)^\epsilon f(v_i) - \tilde{\delta}(v_i, e_q) |e_q| w_I(v_{\tilde{q}})^\alpha \\
&w_G(v_{\tilde{q}})^\eta f(v_{\tilde{q}}) = \\
&\left(\delta(v_i, e_q) w_G(v_i)^\epsilon - \tilde{\delta}(v_i, e_q) |e_q| w_G(v_i)^\eta \right) \\
&w_I(v_i)^\alpha f(v_i)
\end{aligned}$$

With this equality, we obtain the desired result:

$$\begin{aligned}
&\sum_{v_i \in \mathcal{V}} \sum_{e_q \in \mathcal{E}_H} W_I(e_q)^\beta W_G(e_q)^\gamma G(e_q) \\
&\quad \left(\delta(v_i, e_q) w_G(v_i)^\epsilon - \tilde{\delta}(v_i, e_q) |e_q| w_G(v_i)^\eta \right) \\
&= \sum_{v_i \in \mathcal{V}} w_I(v_i)^\alpha f(v_i) \\
&\quad \sum_{e_q \in \mathcal{E}_H} \left(\delta(v_i, e_q) w_G(v_i)^\epsilon - \tilde{\delta}(v_i, e_q) |e_q| w_G(v_i)^\eta \right) \\
&\quad W_I(e_q)^\beta W_G(e_q)^\gamma G(e_q) \\
&= \sum_{v_i \in \mathcal{V}} w_I(v_i)^\alpha f(v_i) \nabla_v^* G(v_i) = \langle f, \nabla_v^* G \rangle_{\mathcal{H}(\mathcal{V})}
\end{aligned}$$

Therefore, the equality $\langle G, \nabla_v f \rangle_{\mathcal{H}(\mathcal{E}_H)} = \langle f, \nabla_v^* G \rangle_{\mathcal{H}(\mathcal{V})}$ holds true for all vertex functions $f \in \mathcal{H}(\mathcal{V})$ and all hyper-edge functions $G \in \mathcal{H}(\mathcal{E}_H)$. \square

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Arnaboldi, V., Conti, M., Passarella, A., Dunbar, R.: Online social networks and information diffusion: The role of ego networks. *Online Social Networks and Media* **1**, 44–55 (2017)
2. Bungert, L., Burger, M.: Asymptotic profiles of nonlinear homogeneous evolution equations of gradient flow type. *Journal of Evolution Equations* **20**, 1061–1092 (2020)
3. Burger, M., He, L., Schönlieb, C.B.: Cahn-Hilliard inpainting and a generalization for grayvalue images. *SIAM Journal on Imaging Sciences* **2**(4), 1129–1167 (2009)
4. Chamley, C., Scaglione, A., Li, L.: Models for the diffusion of beliefs in social networks: An overview. *IEEE Signal Processing Magazine* **30**(3), 16–29 (2013)
5. Di Giovanni, F., Rowbottom, J., Chamberlain, B. P., Markovich, T., Bronstein, M. M.: Graph neural networks as gradient flows. *arXiv preprint arXiv:2206.10991* (2022)
6. Elmoataz, A., Toutain, M., Tenbrinck, D.: On the p-Laplacian and infinity-Laplacian on graphs with applications in image and data

- processing. *SIAM Journal on Imaging Sciences* **8**(4), 2412–2451 (2015)
7. Fazeny, A.: p -Laplacian Operators on Hypergraphs. Master thesis at FAU Erlangen-Nürnberg, https://gitlab.com/arianefazeny/hypergraph_p-laplace/-/raw/main/p-Laplacian_Hypergraphs_Fazeny_Ariane.pdf (2023)
 8. Fazeny, A., Tenbrinck, D., Burger, M.: Hypergraph p -Laplacians, Scale Spaces, and Information Flow in Networks. *Proceedings on 9th International Conference on Scale Space and Variational Methods in Computer Vision*, 677–690 (2023)
 9. Jost, J., Mulas, R., Zhang, D.: p -Laplace Operators for Oriented Hypergraphs. *Vietnam Journal of Mathematics* Oct (2021)
 10. Leskovec J., Krevl A.: SNAP Datasets: Stanford Large Network Dataset Collection, <https://snap.stanford.edu/data/twitter-2010.html>. Last accessed 5 Oct 2022
 11. Li, P., Milenkovic, O.: Submodular hypergraphs: p -laplacians, cheeger inequalities and spectral clustering. *International Conference on Machine Learning*, 3014–3023 (2018)
 12. Majeed, A., Rauf, I.: Graph theory: A comprehensive survey about graph theory applications in computer science and social networks. *Inventions* **5**(1), 10 (2020)
 13. Mulas, R., Kuehn, C., Böhle, T., Jost, J.: Random walks and Laplacians on hypergraphs. *Discrete Applied Mathematics* **317**, 26–41 (2022)
 14. Neuhäuser, L., Lambiotte, R., Schaub, M.: Consensus dynamics and opinion formation on hypergraphs. *Springer International Publishing*, 347–376 (2022)
 15. Solomon, J. PDE approaches to graph analysis. arXiv preprint [arXiv:1505.00185](https://arxiv.org/abs/1505.00185) (2015)
 16. Stanković, L., Daković, M., Sejdić, E.: Introduction to graph signal processing. *Vertex-Frequency Analysis of Graph Signals*, 3–108 (2019)
 17. Turcotte, J., York, C., Irving, J., Scholl, R., Pingree, R.: News recommendations from social media opinion leaders: Effects on media trust and information seeking. *Journal of computer-mediated communication* **20**(5), 520–535 (2015)
 18. Zanette, D.: H: Beyond networks: Opinion formation in triplet-based populations. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **367**, 3311–3319 (2009)
 19. Zhang, S., Ding, Z., Cui, S.: Introducing hypergraph signal processing: Theoretical foundation and practical applications. *IEEE Internet of Things Journal* **7**(1), 639–660 (2019)
 20. Zhou, D., Huang, J., Schölkopf, B.: Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems* **19** (2006). https://proceedings.neurips.cc/paper_files/paper/2006/file/dff8e9c2ac33381546d96deea9922999-Paper.pdf

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.