Quantization of Bandlimited Graph Signals

Felix Krahmer*, He Lyu†, Rayan Saab‡, Anna Veselovska* and Rongrong Wang†

* *Department of Mathematics & Munich Data Science Institute*
Technical University of Munich
and Munich Center for Machine Learning
Garching/Munich, Germany
{felix.krahmer, hanna.veselovska}@tum.de

‡ *Department of Mathematics & Halicioglu Data Science Institute*
University of California, San Diego
La Jolla, San Diego, USA
rsaab@ucsd.edu

† *Department of Computational Mathematics, Science and Engineering & Department of Mathematics*
Michigan State University
East Lansing, USA
{lyuhe, wangron6}@msu.edu

***Abstract*—Graph models and graph-based signals are becoming increasingly important in machine learning, natural sciences, and modern signal processing. In this paper, we address the problem of quantizing bandlimited graph signals. We introduce two classes of noise-shaping algorithms for graph signals that differ in their sampling methodologies. We demonstrate that these algorithms can be efficiently used to construct quantized representatives of bandlimited graph-based signals with bounded amplitude. Moreover, for one of the algorithms, we provide theoretical guarantees on the relative error between the quantized representative and the true signal.**

Key words: bandlimited graph signals, quantization, 1-bit quantization, noise-shaping, graph signal processing

I. State of the art and any preliminary works

Graph signals provide a natural representation of data in many applications, such as social networks, web information analysis, sensor networks and machine learning. Graph signal processing (GSP) is currently an active field of mathematical research that aims to extend the well-developed tools for the analysis of conventional signals to signals on graphs by accounting for the underlying connectivity. A key challenge in graph signal processing is quantization, which involves finding efficient ways to represent the values of a graph signal with a finite number of bits while preserving its information content. In this context, quantizing a "signal" $\boldsymbol{f}$ consists of replacing it by a vector $\boldsymbol{q}$, whose entries are from a finite set (the alphabet), in such a way that a good approximation of $\boldsymbol{f}$ can be subsequently reconstructed from $\boldsymbol{q}$.

Of particular relevance to us are the noise-shaping quantization schemes (see, e.g.,[2]), which share the underlying general approach of placing as much of the difference between the quantized and un-quantized signal in the kernel of the relevant reconstruction operator. Among them, the famous $\Sigma\Delta$ schemes used for quantization of bandlimited functions and images shape the quantization noise by exploiting dependencies between neighboring samples [5, 3, 12, 10]. Among other applications, efficient quantization algorithms have also been developed for compressive sensing measurements [7, 16, 9], and –more recently– post-training quantization for neural networks [18, 6, 13]. Indeed, all these noise-shaping approaches have been shown to outperform their naive counterparts, i.e., those that rely on simply replacing the samples of the signal by their nearest neighbors in the alphabet. Nevertheless, noise-shaping methods have not been proposed for the quantization of graph-based signals due to the challenges associated with accounting for the graph structure.

Inspired by the efficiency of the above-indicated quantization schemes, in this work, we introduce novel noise-shaping quantization techniques for graph signals with a bandlimited spectrum. We propose two types of sampling approaches for quantization. We illustrate that the error emerging after quantization, i.e. quantization noise, has high graph frequency content and, therefore, can be efficiently removed in a suitable reconstruction process. Additionally, we present a theoretical analysis for one of our approaches showing that the quantization error depends on the signal bandwidth and exhibits favorable decay properties with the number of samples.

II. Problem Setting and Notation

We consider an undirected, connected graph $\mathcal{G}=(\mathcal{V},\mathcal{E},W)$ with no self-loops where $\mathcal{V}$ is a set of N vertices, with N assumed to be finite but large, $\mathcal{E}$ is a set of edges, and where W is a weighted adjacency matrix. Let $d_m := \sum_n W_{mn}$ be the degree of the mth vertex. The corresponding normalized Laplacian matrix is defined as $\mathcal{L} = D^{-1/2}(D-W)D^{-1/2}$, where $D = \mathrm{diag}\{d_1, d_2, \cdots, d_N\}$ is the diagonal degree matrix. Here, $\mathcal{L}$ is a symmetric positive semi-definite matrix and has

FK and AV acknowledge support by the German Science Foundation (DFG) in the context of the collaborative research center TR-109 and the Emmy Noether junior research group KR 4512/1-1 as well as by the Munich Data Science Institute and Munich Center for Machine Learning. AV acknowledges the support of the program Global Challenges for Women in Math Science. RS acknowledges support by National Science Foundation Grant DMS-2012546 and a Simons Fellowship. RW acknowledges support by National Science Foundation Grant CCF-2212065.

an orthogonal set of eigenvectors $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$, corresponding to the eigenvalues $\lambda(\mathcal{G}) = \mathrm{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_N\}$ ordered as $\lambda_1 \leq \lambda_2 \leq \cdots \leq \cdots \leq \lambda_N$.

Algorithm 1: Graph Noise Shaping with Permutations

Input: Low-pass graph filter $\boldsymbol{L}$, graph signal $\boldsymbol{f}$, vertex order $\{k_1, \ldots, k_N\}$, number of optimization loops $T \in \mathbb{N}$
Output: Quantized samples $\boldsymbol{q}$, reconstruction $\boldsymbol{f}_q$
Assign: $\boldsymbol{q} \leftarrow$ Algorithm 2
for *epoch = 1:T* **do**
  for *i = 1:N* **do**
    $\boldsymbol{u}_i := \sum_{j \neq k_i} \boldsymbol{\ell}_j (\boldsymbol{f}_j - \boldsymbol{q}_j)$
    $\boldsymbol{q}_{k_i} = Q_\delta\Big(\boldsymbol{f}_{k_i} + \frac{\langle \boldsymbol{\ell}_{k_i}, \boldsymbol{u}_i \rangle}{\|\boldsymbol{\ell}_{k_i}\|_2^2}\Big)$
  end
end
Assign $\boldsymbol{f}_q = \boldsymbol{L}\boldsymbol{q}$

A signal or a function $f : \mathcal{V} \to \mathbb{R}$ defined on vertices of the graph may be represented as a vector $\boldsymbol{f} \in \mathbb{R}^N$, where the nth component of $\boldsymbol{f}$ represents the function value at the nth vertex in $\mathcal{V}$. Generalizing the classical Fourier transform, the eigenvectors and the eigenvalues of matrix $\mathcal{L}$ provide a spectral interpretation of the graph. The eigenvalues $\{\lambda_1, \lambda_2, \cdots, \lambda_N\}$ can be treated as graph frequencies. The eigenvectors of the Laplacian matrix demonstrate increasing oscillatory behavior as the magnitude of the graphs frequencies decreases [17]. The Graph Fourier Transform (GFT) of a signal $\boldsymbol{f}$ is defined as $\widehat{\boldsymbol{f}} = \boldsymbol{X}^T \boldsymbol{f}$ with the entries $\widehat{\boldsymbol{f}}(\lambda_i) = \langle \boldsymbol{f}, \boldsymbol{x}_i \rangle$.

A graph signal $\boldsymbol{f}$ is called bandlimited when there exists $r \in [1, 2, \cdots, N]$ such that its GFT has support only in the frequency interval $[0, \lambda_r]$, see [15, 17]. If we denote the matrix restricted to the first r Fourier vectors by $\boldsymbol{X}_r = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_r]$, then for each r-bandlimited function $\boldsymbol{f}$ there exists a vector $\boldsymbol{\alpha} \in \mathbb{R}^r$ such that $\boldsymbol{f} = \boldsymbol{X}_r \boldsymbol{\alpha}$. Without loss of generality, we consider bandlimited functions normalized to have $\|\boldsymbol{f}\|_\infty \leq 1$.

The graph's geometry and the properties of bandlimited function subspaces are usually defined in terms of graph incoherence [17] as follows.

Definition II.1. [4] For the r-dimensional subspace of graph functions spanned by $\boldsymbol{X}_r = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_r] \in \mathbb{R}^{N \times r}$, let $P_{\boldsymbol{X}_r} = \boldsymbol{X}_r \boldsymbol{X}_r^T$ be the orthogonal projection onto $\boldsymbol{X}_r$. The incoherence of the graph subspace $\boldsymbol{X}_r$ is defined as

$$\mu(\boldsymbol{X}_r) = \frac{N}{r} \max_{1 \leq i \leq N} \|P_{\boldsymbol{X}_r} e_i\|_2^2 . \tag{II.1}$$

For different classes of random graphs, it has been shown that with high probability sufficiently large graphs have small μ [4, 1], which indicates that the graph Laplacian eigenvectors are well spread.

In this work, we are interested in the quantization of bandlimited real-valued graph signals, that is, in representing such graph signals with samples from a certain finite set $\mathcal{A} \subset \mathbb{R}$ which we call an alphabet. To be more precise,

Algorithm 2: Initialization Strategies

Input: Low-pass graph filter $\boldsymbol{L}$, graph signal $\boldsymbol{f}$, order of vertices $\{k_1, \ldots, k_N\}$, weight matrix W, maximal hop-distance s
Output: Quantized samples $\boldsymbol{q}$
if *Step-by-Step-Serving=true* **then**
  Assign $\boldsymbol{q} \leftarrow 0$
  for *i = 1:N* **do**
    $\boldsymbol{q}_{k_i} = Q_\delta\Big(\boldsymbol{f}_{k_i} + \frac{\langle \boldsymbol{\ell}_{k_i}, \boldsymbol{u}_{i-1} \rangle}{\|\boldsymbol{\ell}_{k_i}\|_2^2}\Big)$
    $\boldsymbol{u}_i := \sum_{j=1}^{i-1} \boldsymbol{\ell}_{k_j} (\boldsymbol{f}_{k_j} - \boldsymbol{q}_{k_j}) + \boldsymbol{\ell}_{k_i} (\boldsymbol{f}_{k_i} - \boldsymbol{q}_{k_i})$
  end
end
if *Sigma-Delta-Weights=true* **then**
  The set S holds already quantized vertices
  Initialization: $S \leftarrow \{i\}$, where i is a vertex index satisfying $d_i \geq d_n$, for all $n \in [N]$
  Assign $\boldsymbol{q}_i = Q_\delta(\boldsymbol{f}_i)$ and $\boldsymbol{u}_i = \boldsymbol{f}_i - \boldsymbol{q}_i$;
  while *|S| < N* **do**
    for *k = 1: s* **do**
      $T_k \leftarrow$ index set of k-hop neighbours of v_i
      $P_k \leftarrow$ sort the indices in T_k according to the number of quantized neighbors, i.e., for $j, l \in P_k$, j will be ahead of l if $|\mathcal{N}_{v_l} \cap S| < |\mathcal{N}_{v_j} \cap S|$.
      for *each j in* P_k **do**
        $M_j \leftarrow \mathcal{N}_{v_j} \cap S$
        $\boldsymbol{q}_j = Q_\delta\Big(\boldsymbol{f}_j + \frac{\sum_{k \in M_j} W_{j,k} \boldsymbol{u}_k}{\sum_{k \in M_j} W_{j,k}}\Big)$
        $\boldsymbol{u}_j = \frac{\sum_{k \in M_j} W_{j,k} \boldsymbol{u}_k}{\sum_{k \in M_j} W_{j,k}} + \boldsymbol{f}_j - \boldsymbol{q}_j$
        $S \leftarrow S \cup \{j\}$
      end
    end
  end
end

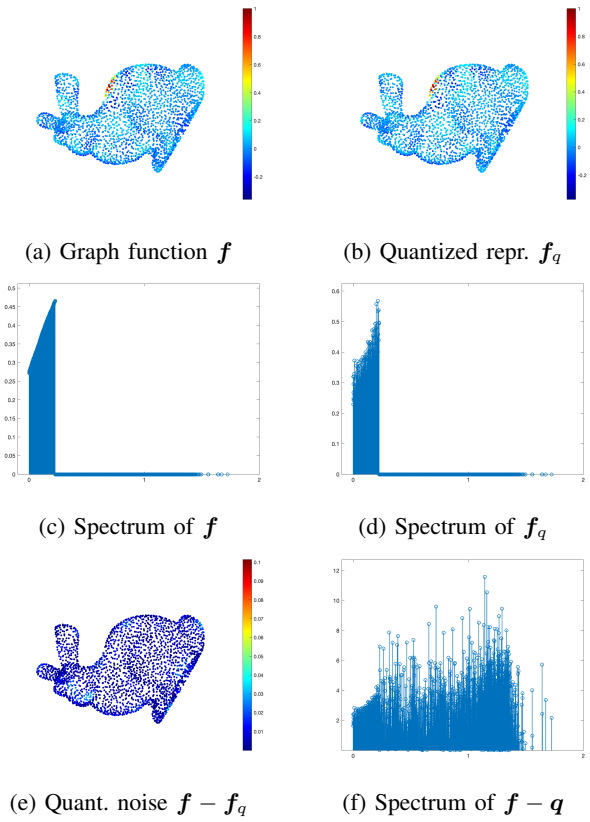
given an r-bandlimited graph signal $\boldsymbol{f} \in \mathbb{R}^N$, our goal is to find a sequence of quantized samples $\boldsymbol{q} \in \mathcal{A}^N$, that is a good representative of $\boldsymbol{f}$ in terms of some quality measure. Motivated by practical application, we aim to find $\boldsymbol{q} \in \mathcal{A}^N$, such that $\boldsymbol{f}$ and $\boldsymbol{q}$ are close to each other under the action of a low-pass graph filter $\boldsymbol{L} := [\boldsymbol{\ell}_1 \ldots, \boldsymbol{\ell}_N] \in \mathbb{R}^{N \times N}$. A quality measure will be fixed to the Euclidean distance, i.e. $\|\boldsymbol{L}(\boldsymbol{f} - \boldsymbol{q})\|_2$.

Moreover, we focus on representing $\boldsymbol{q} \in \mathcal{A}^N$ in terms of the classical midtread alphabet defined as

$$\mathcal{A}_\delta = \mathcal{A}_{\delta,K} := \{\pm k\delta : 0 \leq k \leq K,\, k \in \mathbb{Z}\}, \tag{II.2}$$

where $\delta > 0$ denotes the quantization step size. Additionally, for alphabet $\mathcal{A}_\delta$, we define the associated memoryless scalar quantizer $Q_\delta : \mathbb{R} \to \mathcal{A}_\delta$ as $Q_\delta(z) := \arg\min_{x \in \mathcal{A}_\delta} |z - x|$, that returns the closest element of the alphabet to its argument. As quantized samples $\boldsymbol{q} \in \mathcal{A}_\delta$ will be designed to give a small difference to the original signal $\boldsymbol{f}$ under the low-pass filtering,

we will call $\boldsymbol{f}_q = \boldsymbol{L}\boldsymbol{q}$ a *quantized representative* of $\boldsymbol{f}$ and will refer to $\boldsymbol{L}(\boldsymbol{f}-\boldsymbol{q})$ as the quantization error.



(a) Graph function $\boldsymbol{f}$ (b) Quantized repr. $\boldsymbol{f}_q$

(c) Spectrum of $\boldsymbol{f}$ (d) Spectrum of $\boldsymbol{f}_q$

(e) Quant. noise $\boldsymbol{f}-\boldsymbol{f}_q$ (f) Spectrum of $\boldsymbol{f}-\boldsymbol{q}$

Fig. 1: Performance of the Algorithm. 3 on the bunny graphs of size $N = 2503$ for a graph signal $\boldsymbol{f}$ with bandwidth $r = 100$.

III. Noise-shaping Quantization for Bandlimited Graph Signals with Replacement

In this section, we present the first type of proposed noise-shaping quantization methods for band-limited graph signals. The steps of this graph quantization technique are summarized in Algorithm 1.

Assume that $\boldsymbol{f} \in \mathbb{R}^N$ is an r-bandlimited graph signal and $\boldsymbol{L} := [\boldsymbol{\ell}_1 \dots, \boldsymbol{\ell}_N] \in \mathbb{R}^{N\times N}$ is a low-pass graph filter. Ideally, one would approach the problem of graph signal quantization by solving for $\boldsymbol{q} \in \mathcal{A}_\delta$ that minimizes the difference between $\boldsymbol{f}$ and $\boldsymbol{q}$ under the action of the low-pass filter, i.e., by solving

$$\boldsymbol{q}^\sharp = \arg\min_{\widehat{\boldsymbol{q}}\in\mathcal{A}_\delta^M} \|\boldsymbol{L}(\boldsymbol{f}-\widehat{\boldsymbol{q}})\|_2 . \tag{III.1}$$

Unfortunately, the above problem is an instance of integer least squares, hence NP-hard in general [8]. Instead, inspired by recent methods for neural network quantization [11, 18], we propose iterative methods for choosing the entries of $\boldsymbol{q}$.

To this end, we fix a random permutation of the graph vertices $\mathcal{V}_{perm} = \{k_1, \dots, k_N\}$, and for this fixed order we rewrite the quantization error as a linear combination of the columns of the filter $\boldsymbol{L}(\boldsymbol{f}-\boldsymbol{q}) = \sum_{i=1}^N \boldsymbol{\ell}_{k_i}(\boldsymbol{f}_{k_i} - \boldsymbol{q}_{k_i})$. Then after some initialization of $\boldsymbol{q} \in \mathcal{A}_\delta^N$, for the running index i, we update the value of $\boldsymbol{q}_{k_i}$ by choosing the new quantized sample as

$$\boldsymbol{q}_{k_i} = \arg\min_{\widehat{q}\in\mathcal{A}_\delta} \Big\| \sum_{j\neq k_j} \boldsymbol{\ell}_j(\boldsymbol{f}_j - \boldsymbol{q}_j) + \boldsymbol{\ell}_{k_i}(\boldsymbol{f}_{k_i} - \widehat{q}) \Big\|_2 . \tag{III.2}$$

The above $\boldsymbol{q}_{k_i}$ can be efficiently computed due to the following closed-form solution for the optimization problem,

$$\boldsymbol{q}_{k_i} = Q_\delta\Big(\boldsymbol{f}_{k_i} + \frac{\langle \boldsymbol{\ell}_{k_i}, \boldsymbol{u}_i\rangle}{\|\boldsymbol{\ell}_{k_i}\|_2^2}\Big), \tag{III.3}$$

where the vector $\boldsymbol{u}_i := \sum_{j\neq k_j} \boldsymbol{\ell}_j(\boldsymbol{f}_j - \boldsymbol{q}_j)$ is the *state vector* associated with the iteration. We repeat the above-indicated step for all $i = 1, \dots, N$ to visit all the graph vertices ordered in $\mathcal{V}_{perm}$. Note that this quantization process corresponds to sampling vertices of the graph (or columns of the low-pass filter) at random without replacement and quantizing with respect to the drawn order.

After visiting all graph vertices, there is still a chance that some entries of $\boldsymbol{q}$ can be replaced by other elements of the alphabet and decrease the quantization error. This is due to the greedy local nature of the proposed iteration that cannot be expected to find a global optimum in general. To mitigate the situation, we propose to revisit vertices following the order $\mathcal{V}_{perm}$ several times until $\boldsymbol{q}$ does not update. This corresponds to conducting several loops of the quantization process, the number of which we will denote by $T \in \mathbb{N}$.

In general, the process of updating $\boldsymbol{q}$ in the further loops reaches a stationary point, which we may think of as a local minimum. A good initialization for Algorithm 1 can help us avoid "bad" local minima. Here, we propose two potential approaches, inspired by different quantization techniques (e.g., [5, 11, 12, 10]).

In the first initialization, marked as *Step-by-Step-Serving* in Algorithm 2, for the same fixed vertex order V_{perm} as before, we seek a candidate $\boldsymbol{q}$ which gives potentially small $\|\boldsymbol{L}(\boldsymbol{f}-\boldsymbol{q})\|_2 = \|\sum_{i=1}^N \boldsymbol{\ell}_j(\boldsymbol{f}_j - \widehat{\boldsymbol{q}}_j)\|_2$, by adding the error components column by column. Namely, we start by choosing $\boldsymbol{q}_{k_1}$ which minimizes $\|\boldsymbol{\ell}_{k_1}(\boldsymbol{f}_{k_1} - \boldsymbol{q}_{k_1})\|_2$. In step i, we collect the error made in the previous $i-1$ steps using the state vector $\boldsymbol{u}_{i-1} := \sum_{j=1}^{i-1} \boldsymbol{\ell}_{k_j}(\boldsymbol{f}_{k_j} - \boldsymbol{q}_{k_j})$ and select $\boldsymbol{q}_{k_i}$ as the element in $\mathcal{A}_\delta$ that yields the smallest growth of the error via $\boldsymbol{q}_{k_i} = \arg\min_{\widehat{q}\in\mathcal{A}_\delta} \Big\| \sum_{j=1}^{i-1} \boldsymbol{\ell}_{k_j}(\boldsymbol{f}_{k_j} - \boldsymbol{q}_{k_j}) + \boldsymbol{\ell}_{k_i}(\boldsymbol{f}_{k_i} - \widehat{q}) \Big\|_2$. Here, $\boldsymbol{q}_{k_i}$ can be obtained by a closed-form expression similar to (III.3) with the state $\boldsymbol{u}_i$ changed accordingly. Visiting all N vertices of the graph gives us a reasonable start for the procedure (III.2).

In the second, alternative initialization approach, we traverse the graph using Breadth-First-Search. Starting with the vertex i which has the maximum degree, in step k, we quantize all the k-hop neighbors of vertex i. The vertices are sorted according to the number of quantized neighbors. When quantizing the value of vertex j, we first add to it the weighted sum of all the state variables from its quantized neighbors. This initialization technique is described in detail in Algorithm 2 under the *Sigma-Delta-Weights* choice. Note that in [10, 12], the authors also considered $\Sigma\Delta$ quantization beyond 1D signals, and two

Algorithm 3: Graph Noise Shaping via Step-by-Step Serving with Replacement

Input: Low-pass graph filter $\boldsymbol{L}$, graph signal $\boldsymbol{f}$, number of iterations M

Output: Quant. samples $\boldsymbol{q} \in \widetilde{\mathcal{A}_\delta}^N$, reconstruction $\boldsymbol{f}_q$

Assign $\widetilde{\boldsymbol{q}} \leftarrow 0 \in \mathbb{R}^M$, $\boldsymbol{u}_0 \leftarrow 0 \in \mathbb{R}^N$

for *i= 1:M* **do**

sample uniformly an index $1 \le k_i \le N$

assign $\boldsymbol{v}_i = k_i$

$\widetilde{\boldsymbol{q}}_i = Q_\delta\Big(\boldsymbol{f}_{k_i} + \frac{\langle \boldsymbol{\ell}_{k_i}, \boldsymbol{u}_{i-1}\rangle}{\|\boldsymbol{\ell}_{k_i}\|_2^2}\Big)$

$\boldsymbol{u}_i = \boldsymbol{u}_{i-1} + \boldsymbol{\ell}_{k_i}(\boldsymbol{f}_{k_i} - \widetilde{\boldsymbol{q}}_i)$

end

Assign: $\boldsymbol{q}_i = \sum_{j\in\sigma(i)} \widetilde{\boldsymbol{q}}_j$ and $\boldsymbol{f}_q = \frac{N}{M}\boldsymbol{L}\,\boldsymbol{q}$

weighted $\Sigma\Delta$ approaches were proposed to extend $\Sigma\Delta$ to 2D images. This proposed initialization approach is closely related to the methods in [10, 12] in that they all exploit dependencies between neighboring samples.

We use these two initialization techniques as alternative ways to select starting values of $\boldsymbol{q}$ in Algorithm 1. Our numerical experiments in Section V demonstrate that the two initialization approaches lead to different quantized vectors $\boldsymbol{q} \in \mathcal{A}_\delta^N$. The noise-shaping framework presented in this section exhibits good numerical performance albeit with no theoretical guarantees at the moment.

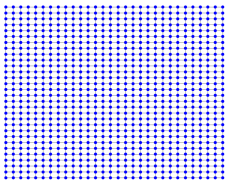
IV. Noise-shaping Quantization with Random Sampling and Theoretical Guarantees

In this section, we propose an alternative noise-shaping algorithm for bandlimited graph signals relying on vertex sampling without replacement. This algorithm shows comparable results with respect to Algorithm 1 in most of the cases, but in contrast to the latter, it allows for qualitative error analysis.

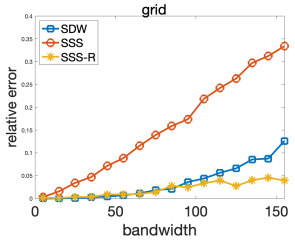
Consider an r-bandlimited graph signal $\boldsymbol{f} \in \mathbb{R}^N$ and a low-pass graph filter $\boldsymbol{L} := [\boldsymbol{\ell}_1 \dots, \boldsymbol{\ell}_N] \in \mathbb{R}^{N\times N}$ and denote by $M \in \mathbb{N}$ the total number of iterations. To find a quantized $\boldsymbol{q}$, in step $1 \le i \le M$, we sample a vertex index k_i uniformly at random from the set of indices $\mathcal{V} = \{1, \dots, N\}$. Then, as before, we quantize $\boldsymbol{f}_{k_i}$ by selecting $\boldsymbol{q}_{k_i} \in \mathcal{A}_\delta$ to minimize the accumulated error. Note that since we sample without replacement, each graph vertex can potentially appear multiple times. Thus, we introduce an auxiliary vector of quantized samples $\widetilde{\boldsymbol{q}} \in \mathcal{A}_\delta^M$ and an index vector $\boldsymbol{v} \in \mathbb{N}^M$ such that

$$\widetilde{\boldsymbol{q}}_i = \arg\min_{\widehat{q}\in\mathcal{A}_\delta} \Big\| \sum_{j=1}^{i-1} \boldsymbol{\ell}_{k_j}(\boldsymbol{f}_{k_j} - \widetilde{\boldsymbol{q}}_j) + \boldsymbol{\ell}_{k_j}(\boldsymbol{f}_{k_i} - \widehat{q}) \Big\|, \quad \text{(IV.1)}$$

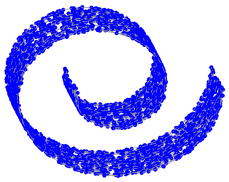
and $\boldsymbol{v}_i = k_i$. At the end of the iteration process, we obtain vector $\widetilde{\boldsymbol{q}} \in \mathcal{A}_\delta^M$ with quantized values, and $\boldsymbol{v} \in \mathbb{N}^M$ with recorded vertex selections. As M can be different from the true dimension N of the signal $\boldsymbol{f}$, in order to obtain the desired quantized samples $\boldsymbol{q} \in \mathcal{A}_\delta^N$ we sum all the entries of $\widetilde{\boldsymbol{q}}$ corresponding to the same vertex and assign them to $\boldsymbol{q}$ via $\boldsymbol{q}_i = \sum_{j\in\sigma(i)} \widetilde{\boldsymbol{q}}_j$ where $\sigma(i) := \{k : \boldsymbol{v}_k = i\}$. This quantization approach is summarized in Algorithm 3. Importantly, the vector $\boldsymbol{q}$ is in general not in $\mathcal{A}_\delta^N$, but it belongs to a slightly larger set generated by an alphabet $\widetilde{\mathcal{A}_\delta}$. Nevertheless, it can be shown that when $M = O(N\log(N))$, $|\widetilde{\mathcal{A}_\delta}| \lesssim |\mathcal{A}_\delta|\log(N)$. The next result shows that $\boldsymbol{f}$ can be well approximated from $\boldsymbol{q} \in \widetilde{\mathcal{A}_\delta}^N$.



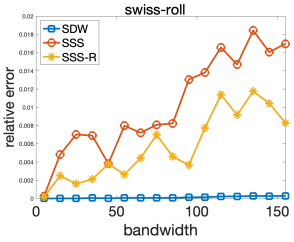
(a) 2D grid graph



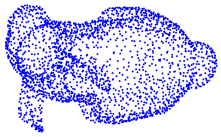
(b) Relative error



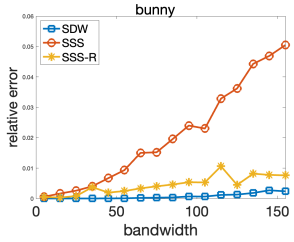
(c) Swiss roll graph



(d) Relative error



(e) Bunny graph



(f) Relative error

Fig. 2: Illustration of performance of the proposed quantization algorithms for bandlimited graph signals on different graphs: Algorithm 1 with two alternative types of initialization: *Step-by-Step-Serving* (SSS) and *Sigma-Delta-Weight* (SDW), and *Step-by-Step-Serving with Replacement* (SSS-R) presented in Algorithm 3.

Theorem IV.1. *Consider a bandlimited graph signal $\boldsymbol{f}$, where $\boldsymbol{f} = \boldsymbol{X}_r\boldsymbol{\alpha}$ for some $\boldsymbol{\alpha} \in \mathbb{R}^r$, with $c \le \|\boldsymbol{f}\|_\infty \le 1$. Assume the K in the definition* (II.2) *of the alphabet $\mathcal{A}_{\delta,K}$ satisfies $K\delta > 1$. In addition, suppose $\boldsymbol{X}_r$ satisfies the incoherence property* (II.1) *with some constant $\mu > 0$. Let $M > 0$ be the number of iterations in Algorithm 3, and $\widetilde{\boldsymbol{q}} \in \widetilde{\mathcal{A}_\delta}^N$ be the resulting quantized vector. Defining $\boldsymbol{f}_q := \frac{N}{M}\boldsymbol{L}\,\boldsymbol{q}$, then with probability at least 1-δ, the relative quantization error satisfies*

$$\frac{\|\boldsymbol{f}_q - \boldsymbol{f}\|_2^2}{\|\boldsymbol{f}\|_2^2} \le C\,\mu^2\,\frac{r^2\log^2(\frac{r}{\delta})}{M},$$

where C is an absolute constant. In addition, $\tilde{\boldsymbol{q}}$ can be represented with $O(N\log\log\frac{N}{\delta})$ bits.

The above-stated results can be obtained following similar ideas as in [18].

V. Numerical Experiments and Conclusion

Here we illustrate the performance of the introduced noise-shaping quantization algorithms on three different graphs and compare their performance in terms of relative signal error.

We consider the grid graph of size $N = 30 \times 30$, the Swiss roll with $N = 3000$ vertices, and the bunny graph which consists of $N = 2503$ vertices, in Fig. 2a-2e. These three graphs have different incoherence properties: the grid is lower than the bunny graph and the Swiss roll is highly coherent [17]. For our illustrations of graph function, we use the GSPBOX toolbox for graph signal processing [14].

For each of the graphs, we construct r-bandlimited signals $\boldsymbol{f}_r = \boldsymbol{X}_r\boldsymbol{\alpha}$ with bandwidth $r \in [5, 155]$ and normalize $\boldsymbol{f}_r$ so that $\|\boldsymbol{f}_r\|_\infty = 1$. We quantize each signal using, first, Algorithm 1 with the two initializations: Step-by-Step-Serving (SSS) and Sigma-Delta-Weight (SDW) choosing $\mathcal{A}_\delta$ of size $\approx \log\log(N)$-bits and $T = 10$, and second, we use Algorithm 3 with the binary alphabet $\mathcal{A} = \{-1, 1\}$ and $M = N\log(N)$ to visit on average all vertices of the graph as in the coupon collector problem. Since in Algorithm 3 storing each element of $\boldsymbol{q}$ will need approximately $\log\log(N)$-bits, this places all the three algorithmic settings on (roughly) equal grounds. To compare the performance of the algorithms, we measure the relative error $\|\boldsymbol{f}_q - \boldsymbol{f}\|_2^2/\|\boldsymbol{f}\|_2^2$. The results of the experiments are depicted in Fig. 2a-2f.

As we can observe, for Algorithm 1 the initialization plays a key role, and using more graph structure for the initial guess, as in SDW initialization, is beneficial for the total performance. Moreover, Algorithm 3 performs better on the graphs with lower incoherence, which is in line with our theoretical findings presented in Theorem IV.1 .

References

[1] Shimon Brooks and Elon Lindenstrauss. "Non-localization of eigenfunctions on large regular graphs". In: *Israel Journal of Mathematics* 193.1 (2013), pp. 1–14.

[2] Evan Chou et al. "Noise-shaping quantization methods for frame-based and compressive sampling systems". In: *Sampling Theory, a Renaissance: Compressive Sensing and Other Developments* (2015), pp. 157–184.

[3] Ingrid Daubechies and Ron DeVore. "Approximating a bandlimited function using very coarsely quantized data: A family of stable sigma-delta modulators of arbitrary order". In: *Annals of mathematics* 158.2 (2003), pp. 679–710.

[4] Yael Dekel, James R Lee, and Nathan Linial. "Eigenvectors of random graphs: Nodal domains". In: *Random Structures & Algorithms* 39.1 (2011), pp. 39–58.

[5] C Sinan Güntürk. "One-bit sigma-delta quantization with exponential accuracy". In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 56.11 (2003), pp. 1608–1630.

[6] C Sinan Güntürk and Weilin Li. "Approximation of functions with one-bit neural networks". In: *arXiv preprint arXiv:2112.09181* (2021).

[7] C Sinan Güntürk et al. "Sobolev duals for random frames and $\Sigma\Delta$ quantization of compressed sensing measurements". In: *Foundations of Computational mathematics* 13 (2013), pp. 1–36.

[8] Babak Hassibi and Haris Vikalo. "On the expected complexity of integer least-squares problems". In: *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 2. IEEE. 2002, pp. II–1497.

[9] Felix Krahmer, Rayan Saab, and Rachel Ward. "Root-exponential accuracy for coarse quantization of finite frame expansions". In: *IEEE transactions on information theory* 58.2 (2012), pp. 1069–1079.

[10] Felix Krahmer and Anna Veselovska. "Enhanced Digital Halftoning via Weighted Sigma-Delta Modulation". In: *arXiv preprint arXiv:2202.04986* (2022).

[11] Eric Lybrand and Rayan Saab. "A greedy algorithm for quantizing neural networks". In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 7007–7044.

[12] He Lyu and Rongrong Wang. "Sigma Delta quantization for images". In: *arXiv preprint arXiv:2005.08487* (2020).

[13] Johannes Maly and Rayan Saab. "A simple approach for quantizing neural networks". In: *arXiv preprint arXiv:2209.03487* (2022).

[14] Nathanaël Perraudin et al. "GSPBOX: A toolbox for signal processing on graphs". In: *arXiv preprint arXiv:1408.5781* (2014).

[15] Isaac Pesenson. "Variational splines and Paley–Wiener spaces on combinatorial graphs". In: *Constructive Approximation* 29 (2009), pp. 1–21.

[16] Rayan Saab, Rongrong Wang, and Özgür Yılmaz. "Quantization of compressive samples with stable and robust recovery". In: *Applied and Computational Harmonic Analysis* 44.1 (2018), pp. 123–143.

[17] David I Shuman, Benjamin Ricaud, and Pierre Vandergheynst. "Vertex-frequency analysis on graphs". In: *Applied and Computational Harmonic Analysis* 40.2 (2016), pp. 260–291.

[18] Jinjie Zhang, Yixuan Zhou, and Rayan Saab. "Post-training quantization for neural networks with provable guarantees". In: *arXiv preprint arXiv:2201.11113* (2022).