Appendix

1 Proof of Lemma 2

Proof. According to the definition, each column vector in the scaled sampling matrix $\mathbf{S} \in \mathbb{R}^{q\times m}$ is sampled without replacement from $\{\mathbf{s}^i = \sqrt{\frac{m}{q}}\mathbf{e}_i^T\}_{i=1}^m \in \mathbb{R}^{1\times m}$ uniformly. Note that

$$\mathbb{E}[\mathbf{X}^T\mathbf{S}^T\mathbf{S}\mathbf{X}] = \mathbb{E}[\mathbf{X}^T\sum_{i=1}^{q}{\mathbf{s}^i}^T\mathbf{s}^i\mathbf{X}] \tag{1}$$

$$= \mathbf{X}^T\sum_{i=1}^{q}\mathbb{E}[{\mathbf{s}^i}^T\mathbf{s}^i]\mathbf{X}. \tag{2}$$

Without loss of generality, we assume that we sample and determine the value of $\{\mathbf{s}^i\}_{i=1}^q$ in the order of the smallest to the largest index. Then, due to that

$$\mathbb{P}(\mathbf{s}^i = \sqrt{\frac{m}{q}}\mathbf{e}_k^T) \tag{3}$$

$$= \mathbb{P}(\mathbf{s}^i = \sqrt{\frac{m}{q}}\mathbf{e}_k^T | \mathbf{s}^j \neq \sqrt{\frac{m}{q}}\mathbf{e}_k^T, \forall j \in [i-1])$$

$$* \mathbb{P}(\mathbf{s}^j \neq \sqrt{\frac{m}{q}}\mathbf{e}_k^T, \forall j \in [i-1]) \tag{4}$$

$$= \frac{1}{m-(i-1)}\frac{m-(i-1)}{m-(i-2)}\cdots\frac{m-1}{m} = \frac{1}{m}, \tag{5}$$

we have

$$\mathbb{E}[{\mathbf{s}^i}^T\mathbf{s}^i] = \sum_{k=1}^{m}\mathbb{P}(\mathbf{s}^i = \sqrt{\frac{m}{q}}\mathbf{e}_k^T)\frac{m}{q}\mathbf{e}_k\mathbf{e}_k^T \tag{6}$$

$$= \sum_{k=1}^{m}\frac{1}{m}\frac{m}{q}\mathbf{e}_k\mathbf{e}_k^T = \frac{1}{q}\mathbf{I}_m. \tag{7}$$

Thus, substituting Eq. (7) into Eq. (2) concludes the proof. □

2 More Comparisons with Batch Solutions

We also compare OSH and our FROSH against two leading batch methods SGH [3] and OCH [7]. Based on the literature [3, 7, 9], we know that SGH maintains a superior tradeoff between the learning accuracy and the scalability, and OCH is the state-of-the-art regarding the accuracy performance compared with the other unsupervised hashing methods such as SpH [11], AGH [10], IsoH [4], DGH [9], and OEH [8].

Distinguished with the bath methods, OSH and FROSH are able to adapt the hash functions to the new coming data. Besides, FROSH enjoys superior training efficiency, i.e., *single pass, lowest costs of space ($O(\ell d)$) and time*, while the unsupervised batch methods such as SGH and OCH require either $O(nd)$ space or multiple passes over the data to load all data into memory or both. Regarding the space cost, OSH and FROSH can throw away the used data and update only on the newly coming data via $O(\ell d)$ space while SGH and OCH should maintain a large extern storage to keep all observed data and the newly coming data. Moreover, given the 19GB data FLICKR-25600 that is merely a small subset of the entire FLICKR image collection, if avoiding multiple passes is required, both the SGH and OCH methods have to maintain the entire FLICKR-25600 data in the memory and keep the intermediate computational results (can be several times larger than FLICKR-25600 itself) in the memory as well, which is infeasible for common computers.

In Table 2, we also provide the empirical time comparisons because it is not clear enough to directly compare the associated time complexities when all methods have different parameters. Note that the released OCH codes have been highly optimized by the authors in contrast to the version used in its original paper [7]. We report the accumulated time after all rounds for OSH and FROSH and provide the time consumptions of training SGH and OCH only on all observed data. Overall, our FROSH offers about $10 \sim 70$ times speed-up than the other com-

Table 2: Comparisons of the training time (in sec.). We report the accumulated time after all rounds for OSH and FROSH and provide the time consumptions of training SGH and OCH only on all observed data.

Dataset	Method	32bits	64bits	128bits
CIFAR-10	SGH	7.83	11.35	19.49
	OCH	26.89	26.95	27.49
	OSH	7.78	11.88	22.09
	FROSH	**0.63**	**0.94**	**2.11**
MNIST	SGH	10.47	14.59	23.47
	OCH	40.45	40.49	41.10
	OSH	13.25	18.93	30.75
	FROSH	**1.17**	**1.49**	**2.56**
GIST-1M	SGH	231	275	290
	OCH	1042	1089	1192
	OSH	228	331	520
	FROSH	**21**	**27**	**45**
FLICKR-25600	SGH	3032	3541	4903
	OCH	4981	5300	5441
	OSH	679	1283	2570
	FROSH	**72**	**92**	**134**

pared solutions. If considering the case where batch-learners should repeatedly do batch learning on both the newly coming data and the currently observed data, SGH and OCH would require significantly more training time than that in Table 2.

3 Approximation for the Projection Matrix W

In this part, we offer more details to show how the project matrix $\mathbf{W}^T \in \mathbb{R}^{r\times d}$ can be approximated. We let $m = \Theta(d)$, and assume $n = \Omega(\ell^{3/2}d^{3/2})$ for simplicity, then the error bound of Eq. (5) in Theorem 2 of the main text becomes $\widetilde{O}(\frac{1}{\ell}\|(\mathbf{A}-\boldsymbol{\mu})\|_F^2)$. Based on it, we give Theorem 3.

Theorem 3. *Given data $\mathbf{A} \in \mathbb{R}^{n\times d}$ with its row mean vector $\boldsymbol{\mu} \in \mathbb{R}^{1\times d}$, let the sketching matrix $\mathbf{B}^{\ell\times d}$ be generated by FROSH in Algorithm 4. Let $m = \Theta(d)$, and assume $n = \Omega(\ell^{3/2}d^{3/2})$ for simplicity. Given $(\mathbf{A}-\boldsymbol{\mu}) \in \mathbb{R}^{n\times d}$ that means subtracting each row of $\mathbf{A}$ by $\boldsymbol{\mu}$, let $h = \|(\mathbf{A}-\boldsymbol{\mu})\|_F^2/\|(\mathbf{A}-\boldsymbol{\mu})\|_2^2$ and σ_i be the i-th largest singular value of $(\mathbf{A}-\boldsymbol{\mu})$. If the sketching size $\ell = \Omega(\frac{h\sigma_1^2}{\epsilon\sigma_{r+1}^2})$, then with probability defined in Theorem 1 we have*

$$\begin{aligned}&\|(\mathbf{A}-\boldsymbol{\mu}) - (\mathbf{A}-\boldsymbol{\mu})\mathbf{W}_{\mathbf{B}}\mathbf{W}_{\mathbf{B}}^T\|_2^2\\ &\le (1+\epsilon)\|(\mathbf{A}-\boldsymbol{\mu}) - (\mathbf{A}-\boldsymbol{\mu})\mathbf{W}\mathbf{W}^T\|_2^2, \quad (8)\end{aligned}$$

where $0 < \epsilon < 1$, $\mathbf{W}_{\mathbf{B}}^T \in \mathbb{R}^{r\times d}$ contains the top r right singular vectors of $\mathbf{B}^{\ell\times d}$, and $\mathbf{W}^T \in \mathbb{R}^{r\times d}$ contains the top r right singular vectors of $(\mathbf{A}-\boldsymbol{\mu}) \in \mathbb{R}^{n\times d}$.

Remark. The bound on $\|(\mathbf{A}-\boldsymbol{\mu}) - (\mathbf{A}-\boldsymbol{\mu})\mathbf{W}_{\mathbf{B}}\mathbf{W}_{\mathbf{B}}^T\|_2^2$ shows the similarity between $\mathbf{W}_{\mathbf{B}}\mathbf{W}_{\mathbf{B}}^T$ and $\mathbf{W}\mathbf{W}^T$. If $\epsilon = 0$, we will have $\mathbf{W}_{\mathbf{B}}\mathbf{W}_{\mathbf{B}}^T = \mathbf{W}\mathbf{W}^T$. However, it cannot characterize the similarity between $\mathbf{W}_{\mathbf{B}} \in \mathbb{R}^{d\times r}$ and $\mathbf{W} \in \mathbb{R}^{d\times r}$, because Eq. (8) of Theorem 3 may also indicate that $\mathbf{W}_{\mathbf{B}}$ approximates $\mathbf{W}\boldsymbol{\Upsilon}$, where $\boldsymbol{\Upsilon} \in \mathbb{R}^{r\times r}$ is an arbitrary unitary matrix with $\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T = \mathbf{I}_r$ and $\mathbf{I}_r$ being an identity matrix so that $\mathbf{W}\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T\mathbf{W}^T = \mathbf{W}\mathbf{W}^T$. Fortunately, due to that $\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T = \mathbf{I}_r$ (i.e., $\boldsymbol{\Upsilon} \in \mathbb{R}^{r\times r}$ is an orthogonal rotation), $\mathbf{W}\boldsymbol{\Upsilon}$ will still retain all information of $\mathbf{W}$ and even empirically get better hashing accuracy, which has been mentioned in Remark 1 of the main text. Therefore, Theorem 3 shows how $\mathbf{W}_{\mathbf{B}}$ approximates $\mathbf{W}$ or $\mathbf{W}\boldsymbol{\Upsilon}$, which can be used to show the effectiveness of the related hashing algorithm.

Here, we restate Remark 1 of the main text: To address the problem that most of the information can be contained by only a small number of significant singular vectors in $\mathbf{W} \in \mathbb{R}^{d\times r}$, OSH [5] also empirically applies a random orthogonal rotation $\boldsymbol{\Upsilon} \in \mathbb{R}^{r\times r}$ (the orthonormal bases of an $r \times r$ random Gaussian matrix) to all singular vectors $\mathbf{W} \in \mathbb{R}^{d\times r}$ returned by Algorithm 1 via $\mathbf{W}\boldsymbol{\Upsilon}$. This step resembles Iterative Quantization [2] but runs much more efficiently with streaming settings maintained and negligible computational cost incurred. Thus, following OSH, our method FROSH also applies $\boldsymbol{\Upsilon} \in \mathbb{R}^{r\times r}$ to the obtained top r right singular vectors of $\mathbf{B}^{\ell\times d}$.

3.1 Proof of Theorem 3

The proof follows by combining our proposed Theorem 2 and Section 1.4 of [6].

Proof. Since $\mathbf{W}_{\mathbf{B}}^T \in \mathbb{R}^{r\times d}$ contains the top r right singular vectors of $\mathbf{B}^{\ell\times d}$, we have $\mathbf{W}_{\mathbf{B}}\mathbf{W}_{\mathbf{B}}^T \in \mathbb{R}^{d\times d}$ as the projection matrix of $\mathbf{B}^{\ell\times d}$. Via Lemma 4 in [1], we have

$$\begin{aligned}&\|(\mathbf{A}-\boldsymbol{\mu}) - (\mathbf{A}-\boldsymbol{\mu})\mathbf{W}_{\mathbf{B}}\mathbf{W}_{\mathbf{B}}^T\|_2^2\\ &\le \sigma_{r+1}^2 + 2\|(\mathbf{A}-\boldsymbol{\mu})^T(\mathbf{A}-\boldsymbol{\mu}) - \mathbf{B}^T\mathbf{B}\|_2, \quad (9)\end{aligned}$$

where σ_i is the i-th largest singular value of $(\mathbf{A}-\boldsymbol{\mu})$.

For simplicity, when $m = \Theta(d)$ and $n = \Omega(\ell^{3/2}d^{3/2})$, the error bound of Eq. (5) in Theorem 2 of the main text will become $\widetilde{O}(\frac{1}{\ell}\|(\mathbf{A}-\boldsymbol{\mu})\|_F^2)$, which is then incorporated into Eq. (9) to get that

$$\begin{aligned}&\|(\mathbf{A}-\boldsymbol{\mu}) - (\mathbf{A}-\boldsymbol{\mu})\mathbf{W}_{\mathbf{B}}\mathbf{W}_{\mathbf{B}}^T\|_2^2\\ &\le \sigma_{r+1}^2 + \widetilde{O}(\frac{1}{\ell}\|(\mathbf{A}-\boldsymbol{\mu})\|_F^2). \quad (10)\end{aligned}$$

Let $h = \|(\mathbf{A}-\boldsymbol{\mu})\|_F^2/\|(\mathbf{A}-\boldsymbol{\mu})\|_2^2$ be the numeric rank of $(\mathbf{A}-\boldsymbol{\mu}) \in \mathbb{R}^{n\times d}$, which could be much smaller than d for a low-rank matrix $(\mathbf{A}-\boldsymbol{\mu}) \in \mathbb{R}^{n\times d}$ with $d < n$. If $\ell = \Omega(\frac{h\sigma_1^2}{\epsilon\sigma_{r+1}^2})$, then from Eq. (10) we have

$$\begin{aligned}&\|(\mathbf{A}-\boldsymbol{\mu}) - (\mathbf{A}-\boldsymbol{\mu})\mathbf{W_B}\mathbf{W_B}^T\|_2^2 \leq (1+\epsilon)\sigma_{r+1}^2 \\ &= (1+\epsilon)\|(\mathbf{A}-\boldsymbol{\mu}) - (\mathbf{A}-\boldsymbol{\mu})\mathbf{W}\mathbf{W}^T\|_2^2, \qquad (11)\end{aligned}$$

where $\sigma_1^2 = \|(\mathbf{A}-\boldsymbol{\mu})\|_2^2$ and $\sigma_{r+1}^2 = \|(\mathbf{A}-\boldsymbol{\mu}) - (\mathbf{A}-\boldsymbol{\mu})\mathbf{W}\mathbf{W}^T\|_2^2$ according to the definition. □

References

[1] P. Drineas and R. Kannan. Pass efficient algorithms for approximating large matrices. In *SODA*, volume 3, pages 223–232, 2003.

[2] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2916–2929, 2013.

[3] Q.-Y. Jiang and W.-J. Li. Scalable graph hashing with feature transformation. In *IJCAI*, pages 2248–2254, 2015.

[4] W. Kong and W.-J. Li. Isotropic hashing. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2012.

[5] C. Leng, J. Wu, J. Cheng, X. Bai, and H. Lu. Online sketching hashing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2503–2511, 2015.

[6] E. Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 581–588. ACM, 2013.

[7] H. Liu, R. Ji, Y. Wu, and F. Huang. Ordinal constrained binary code learning for nearest neighbor search. In *AAAI*, 2017.

[8] H. Liu, R. Ji, Y. Wu, and W. Liu. Towards optimal binary code learning via ordinal embedding. In *AAAI*, pages 1258–1265, 2016.

[9] W. Liu, C. Mu, S. Kumar, and S.-F. Chang. Discrete graph hashing. In *Advances in Neural Information Processing Systems*, pages 3419–3427, 2014.

[10] W. Liu, J. Wang, S. Kumar, and S.-F. Chang. Hashing with graphs. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1–8, 2011.

[11] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in neural information processing systems*, pages 1753–1760, 2009.