A NEED FOR RANGE-NET

To illustrate the limitations of current streaming randomized SVD approaches, we consider a synthetic data matrix X with slow decay in the singular value. The numerical results section later presents a number of these singular value spectra for different practical datasets (**Fig. 3**) to demonstrate that the decay rates are subjective to the problem at hand.

$$X = \text{diag}(\underbrace{450, 449, \cdots, 2, 1}_{f=450}, \underbrace{0, \cdots, 0}_{n-f})$$

Here $X \in \mathbb{R}^{m \times n}$ is a strictly diagonal matrix with $m = n = 500$ with rank $f = 450$, where the singular value spectrum decays linearly. **Fig. 8** shows a comparison of reconstruction errors (see Metrics in **Appendix E.5** for the definition) for SketchySVD (Tropp et al., 2019) (red line), Block Lanczos with Power Iteration (Musco & Musco, 2015) (black line), Sklearn's randomized SVD (skr) implementation (Halko et al., 2011) with (solid cyan line) and without power iteration (dashed blue line), and Range-Net (green line) over 1000 runs for this synthetic dataset.

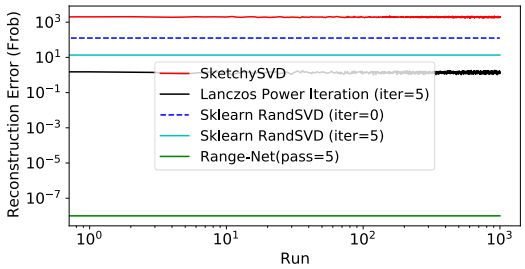
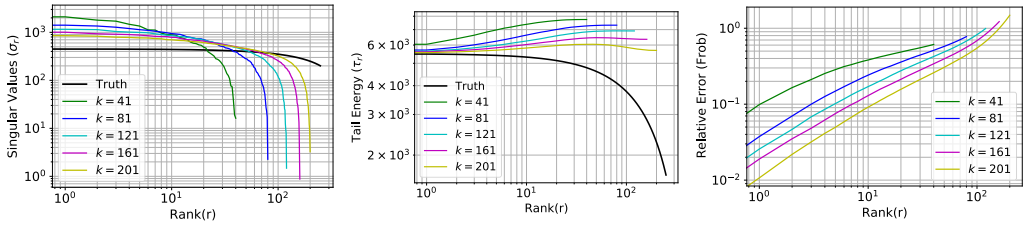


Figure 8: Reconstruction errors ($r = 20$) for Range-Net and randomized SVD schemes (with and without power iterations) for the non-exponentially decaying singular value spectrum over 1000 runs.

Please note that although power iteration improves the reconstruction error for both Block Lanczos (Musco & Musco, 2015) and Sklearn's RandSVD (Halko et al., 2011), power iteration itself requires a persistent presence of the data matrix X in the main memory. For a practical big data scenario, power iteration is therefore not a feasible alternative when the data matrix X or it's sketch is itself too big to be loaded into the main memory. *Note that the error expectation (upper bound) over multiple runs of Randomized SVD algorithms does not reduce.* We further identify the following requirements for SketchySVD to return SVD factors with relatively lower approximation errors:

1. Decay rate of singular values of a dataset must be exponential.
2. For a rank-f matrix, the desired rank r must be chosen such that the oversampled rank k is strictly greater than f ($k \geq f$) to achieve lower errors at scale compared to other runs.

We suggest that the reader also attempt the case where all the diagonal entries are strictly ones and zeros under a high rank setting.



(a) Estimated Singular Values (b) Tail Energy (c) Relative Tail Energy

Figure 9: SketchySVD approximation errors for a synthetic dataset with linear decay in singular value spectrum, corresponding to $r = [10, 20, 30, 40, 50]$ with corresponding oversampled rank $k = [41, 81, 121, 161, 201]$. Since the decay is non-exponential, SketchySVD accrues large approximation errors, hence impractical for real datasets with similar behavior.

Fig. 9 shows the singular values extracted by SketchySVD for a linearly decaying spectrum with corresponding errors in absolute and relative tail energies. The reader is referred to **Appendix**

B.1 for the definitions of tail energy and relative tail energy. Note that the synthetic data is a diagonal matrix chosen specifically so that the exact tail energies can be computed using Frobenius norm as $\left(\sum_{i=r+1}^{k} x_{ii}^2\right)^{\frac{1}{2}}$. For a rank-$r$ approximation, SketchySVD suggests oversampling by a factor of $k = 4r + 1$ to extract the rank-r factors correctly. Hence for an oversampled rank $k = [41, 81, 121, 161, 201]$ the corresponding top rank $r = [10, 20, 30, 40, 50]$ extracted singular values and vectors will have the lowest approximation errors. However as shown in the **Fig. 9 (a)** the extracted singular values have an order of magnitude difference *w.r.t.* the ground truth. Consequently, **Fig. 9 (b)** shows that the absolute tail-energies of the extracted features deviate quite substantially from the true tail-energy. Furthermore, we also notice that the deviations remains large as long as the oversampled rank-k is such that $k < f$. For a practical dataset f is either unknown or almost full rank $f \leq \min(m, n)$ or both and can only be detected by performing a full SVD on the dataset. This poses a serious restriction on SketchySVD's reliability for a realistic big data application, due to an exponential decay assumption.

We also notice that for smaller values of r, the accrued error in both the extracted singular values and tail energy error is worse. **Fig. 9 (b)** shows that for different rank approximations SketchySVD tail energy deviates from the truth quite substantially. This is due to the fact that the oversampled rank $k = 4r + 1 \ll f$, as pointed out before. Note that for oversampling parameter $k < f$, although the error decreases as $k \to f$ the memory requirement increases correspondingly as $\mathcal{O}(k)$ for extracting a low rank-r approximation. This implies that for slow decaying spectrum optimal values of k are such that $k \geq f$ even when a rank-1 approximation is desired. In **Fig. 9 (c)**, one can observe *relative errors* between 10^{-2} and 10^{0}. Although this implies that the actual rank-r tail-energy approximation error is off by 1% at the best, the extracted singular values and vectors are off by one order of magnitude. As a consequence, the extracted singular vectors no longer represent the features of the dataset.

Remark. *For a randomized SVD algorithm to converge (without power iterations) to a rank-r approximation X_r over multiple runs, we posit that a rank-r sketch matrix $\hat{X}$ for a given rank-f dataset X, for $f \geq r$, be such that $P(span\{\hat{X}\} \cap span\{X_r\} = span\{X_r\}) \geq 0.5$. However, ensuring this requires substantial amount of prior knowledge or intelligent sampling (a multipass iterative algorithm).*

Range-Net with it's explicit minimization of tail energy is capable of intelligent sampling on an arbitrary matrix without requiring any prior information. The key point to note here is that Range-Net relies upon an iterative computation of a near optimal projector instead of arbitrary/user-specified projectors used in Randomized SVD schemes. Even if the tail energy is theoretically upper bounded for some of the Randomized SVD schemes, the target is to find the lower-bound (minimizer) on the tail energy as discussed in **Section 1.1**. Furthermore, since none of the randomized SVD schemes construct the projector in an iterative manner while minimizing Eq. 1, the relative error in the tail energy remains high. Even if multiple runs of SketchySVD or Sklearn's RandSVD are performed, the reconstruction errors in tail energies remain the same at scales shown in **Fig. 8**. We would also like to point out that although one must strive for lower errors (relative or otherwise) and tighter theoretical upper and lower bounds, in practice we should also closely monitor if these theoretical bounds deliver us the desired solution.

B Theoretical Guarantees

B.1 Preliminaries

The Frobenius norm of a matrix A is given by,

$$\|A\|_F = \left(\sum_i \sum_j a_{ij}^2\right)^{\frac{1}{2}} = \left(\mathrm{Tr}(A^T A)\right)^{\frac{1}{2}} = \left(\mathrm{Tr}(AA^T)\right)^{\frac{1}{2}}$$

Further, the Frobenius norm can be used to bound Trefethen & Bau III (1997) the norm of a matrix product as,

$$\|AB\|_F \leq \|A\|_F \|B\|_F$$

For a Frobenius norm we have that,

$$\|A + B\|_F \leq \|A\|_F + \|B\|_F$$
$$\|A - B\|_F \geq \big|\|A\|_F - \|B\|_F\big|$$

Also the Frobenius norm of a rank-f matrix A,

$$\|A\|_F = \|\Sigma_f\|_F$$

where $\Sigma_f = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_f)$ and σ_is are the f non-zero, singular values of A.

Let A, B, C be matrices such that the following matrix products are feasible. The cyclic property of the linear trace operator is,

$$\text{Tr}(ABC) = \text{Tr}(BCA) = \text{Tr}(CAB)$$

Definition 1. *The tail energy of an arbitrary matrix $B \in \mathbb{R}^{m\times n}$ with respect to a given matrix $X \in \mathbb{R}^{m\times n}$ equipped with a Frobenius norm is defined as,*

$$\tau = \|X - B\|_F$$

Definition 2. *Let $r, f \in \mathbb{Z}^+$ be positive integers such that $0 < r \leq f$, then a rank-r truncation X_r of a rank-f matrix X is defined as,*

$$X_r = U_r \Sigma_r V_r^T = X V_r V_r^T$$

where, $\Sigma_r = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_r)$ and σ_is are the top r singular values of X and $V_r = [v_1, v_2, \cdots, v_r]$, and $U_r = [u_1, u_2, \cdots, u_r]$ are matrices such that v_is and u_is are the corresponding right and left singular vectors, respectively.

The relative tail energy of a rank-r matrix B_r with respect to a rank-f matrix X ($r \leq f$) is then defined as,

$$\tau_{rel,r} = \frac{\|X - B_r\|_F}{\|X - X_r\|_F} - 1$$

B.2 Stage 1

Proof of Theorem 2. *For any $r, f \in \mathbb{Z}^+$, $0 < r \leq f$, if the tail energy of a rank-f matrix $X \in \mathbb{R}^{m\times n}$, $f \leq min(m, n)$, with respect to an arbitrary rank-r matrix $B_r = X\tilde{V}_r\tilde{V}_r^T$ is bounded below by the tail energy of X with respect to it's rank-r approximation $X_r = XV_rV_r^T$ as,*

$$\|X - B_r\|_F \geq \|X - X_r\|_F$$

where, $V_r = span\{v_1, v_2, \cdots, v_r\}$ and v_is are the right singular vectors corresponding to the largest r singular values then the minimizer of $\underset{\tilde{V}\in\mathbb{R}^{(n\times r)}}{\arg\min} \|X - X\tilde{V}\tilde{V}^T\|_F$ is V_* such that $V_*V_*^T = V_rV_r^T$.

From **Theorem 1** we have,

$$\|X - B_r\|_F - \|X - X_r\|_F \geq 0 \tag{4}$$

Let $V_r = \{v_1, v_2, \ldots, v_r\}$ be the top-r, right-singular vectors of X corresponding to the largest singular values,

$$X_r = XV_rV_r^T = U\Sigma V^T V_r V_r^T = U\Sigma V_r^T \tag{5}$$

Also let $B_r = X\tilde{V}_r\tilde{V}_r^T$ where $\tilde{V}_r$ is an arbitrary rank-r matrix. From triangle inequality we have that,

$$\|X(V_rV_r^T - \tilde{V}_r\tilde{V}_r^T)\|_F \geq \|X(I_n - \tilde{V}_r\tilde{V}_r^T)\|_F - \|X(I_n - V_rV_r^T)\|_F \tag{6}$$

Combining Eq. 4 and Eq. 6 we get,

$$\|X(V_rV_r^T - \tilde{V}_r\tilde{V}_r^T)\|_F \geq 0 \tag{7}$$

Additionally,

$$\|X\|_F\|V_rV_r^T - \tilde{V}_r\tilde{V}_r^T\|_F \geq \|X(V_rV_r^T - \tilde{V}_r\tilde{V}_r^T)\|_F \tag{8}$$

Using the above two inequalities we arrive at,

$$\|X\|_F\|V_rV_r^T - \tilde{V}_r\tilde{V}_r^T\|_F \geq 0 \tag{9}$$

Since $\|X\|_F > 0$, equality is achieved when $\tilde{V}_r\tilde{V}_r^T = V_*V_*^T = V_rV_r^T$. In other words, $span\{V_r\} = span\{V_*\}$ since $(V_*\Theta_r)(V_*\Theta_r)^T = V_*V_*^T$ for any rank-r, real valued, unitary matrix Θ_r spanning the top rank-r subspace of X.

Remark. *Theorem 2 also implies that any matrix $\tilde{V}_r\tilde{V}_r^T$ that does not span the same rank-r subspace of X as $V_rV_r^T$ will result in a higher tail-energy than given by the EYM tail-energy bound equipped with a Frobenius norm.*

Proof of Lemma 2.1. *If $V_r^TV_r = I_r$ and $V_rV_r^T = V_*V_*^T$ then $V_*^TV_* = I_r$.*

Let us assume $V_r^TV_r = I_r$ then,

$$\|V_r^TV_r - I_r\|_F^2 = 0$$
$$\text{Tr}\left(V_r^TV_rV_r^TV_r\right) + \text{Tr}\left(I_r\right) - 2\,\text{Tr}\left(V_r^TV_r\right) = 0$$

Using the cyclic property of the trace operator we have,

$$\text{Tr}\left(V_rV_r^TV_rV_r^T\right) + \text{Tr}\left(I_r\right) - 2\,\text{Tr}\left(V_rV_r^T\right) = 0$$

Using **Theorem 2**, $V_*V_*^T = V_rV_r^T$,

$$\text{Tr}\left(V_*V_*^TV_*V_*^T\right) + \text{Tr}\left(I_r\right) - 2\,\text{Tr}\left(V_*V_*^T\right) = 0$$

Again using the cyclic property of the trace operator we now get,

$$\text{Tr}\left(V_*^TV_*V_*^TV_*\right) + \text{Tr}\left(I_r\right) - 2\,\text{Tr}\left(V_*^TV_*\right) = 0$$

Hence, $V_*^TV_* = I_r$. This shows that following **Theorem 2**, the matrix V_* comprises of orthonormal column vectors spanning the same top rank-r subspace of X as the orthonormal column vectors V_r.

Proof of Lemma 2.2. *If $X \in \mathbb{R}^{m\times n}$ is a rank f matrix, then for any rank $r > f$, where $\{r, f\} \leq \min(m, n)$, if $V_*^TV_* = I_r$ and $V_*V_*^T = V_rV_r^T$ then $V_r^TV_r = I_r$.*

$$\|V_*^TV_* - I_r\|_F^2 = 0$$
$$\text{Tr}\left(V_*^TV_*V_*^TV_*\right) + \text{Tr}\left(I_r\right) - 2\,\text{Tr}\left(V_*^TV_*\right) = 0$$

Using the cyclic property of the trace operator we have,

$$\text{Tr}\left(V_*V_*^TV_*V_*^T\right) + \text{Tr}\left(I_r\right) - 2\,\text{Tr}\left(V_*V_*^T\right) = 0$$

Using **Theorem 2**, $V_*V_*^T = V_rV_r^T$,

$$\text{Tr}\left(V_rV_r^TV_rV_r^T\right) + \text{Tr}\left(I_r\right) - 2\,\text{Tr}\left(V_rV_r^T\right) = 0$$

Again using the cyclic property of the trace operator,

$$\text{Tr}\left(V_r^TV_rV_r^TV_r\right) + \text{Tr}\left(I_r\right) - 2\,\text{Tr}\left(V_r^TV_r\right) = 0$$
$$\text{Tr}\left(V_r^TV_rV_r^TV_r + I_r - 2V_r^TV_r\right) = 0$$
$$\|V_r^TV_r - I_r\|_F = 0$$

Hence $V_r^TV_r = I_r$.

Remark. *Lemma 2.2 shows that for a rank-r approximation of a rank-f matrix X such that $r > f$, the extracted right singular vectors are orthonormal when $V_*^TV_* = I_r$. This justifies the constraint $\tilde{V}^T\tilde{V} = I_r$ for the stage-1 minimization problem in Eq. 2 and is numerically verified in **Appendix D**.*

Remark. *Note that Lemma 2.1 and 2.2 does not imply that $V_* = V_r$ instead $V_*\Theta_r = V_r$, where Θ_r is any real valued unitary matrix for the equality to hold true.*

B.3 STAGE 2

Proof of Theorem 3. *Given a rank-r matrix $XV_* \in \mathbb{R}^{m\times r}$ and an arbitrary, rank-r matrix $C \in \mathbb{R}^{m\times r}$, following **Theorem 1**, the tail energy of XV_* with respect to XV_*C is bounded as,*

$$\|XV_* - XV_*C\|_F \geq 0$$

where the equality holds true if and only if $C = I_r$.

$$\begin{aligned}
\|XV_*(I_r - C)\|_F &\geq 0 \\
\|XV_*\|_F\|I_r - C\|_F &\geq \|XV_*(I_r - C)\|_F \\
\|XV_*\|_F\|I_r - C\|_F &\geq 0
\end{aligned}$$

Since $XV_* > 0$, this implies equality is achieved if and only if $C = I_r$.

Proof of Lemma 3.1 *If $C = \Theta_r\Theta_r^T$, where $\Theta_r \in \mathbb{R}^{r\times r}$ is a rank-r matrix such that $C = I_r$, then Θ_r is a real-valued unitary matrix in an r-dimensional Euclidean space.*

$$\begin{aligned}
\Theta_r\Theta_r^T &= I_r \\
\|\Theta_r\Theta_r^T - I_r\|_F^2 &= 0 \\
\operatorname{Tr}\left(\Theta_r\Theta_r^T\Theta_r\Theta_r^T + I_r - 2\Theta_r\Theta_r^T\right) &= 0 \\
\operatorname{Tr}\left(\Theta_r\Theta_r^T\Theta_r\Theta_r^T\right) + \operatorname{Tr}\left(I_r\right) - 2\operatorname{Tr}\left(\Theta_r\Theta_r^T\right) &= 0
\end{aligned}$$

Using the cyclic property of the trace operator,

$$\begin{aligned}
\operatorname{Tr}\left(\Theta_r^T\Theta_r\Theta_r^T\Theta_r\right) + \operatorname{Tr}\left(I_r\right) - 2\operatorname{Tr}\left(\Theta_r^T\Theta_r\right) &= 0 \\
\operatorname{Tr}\left((\Theta_r^T\Theta_r - I_r)(\Theta_r^T\Theta_r - I_r)^T\right) &= 0 \\
\|\Theta_r^T\Theta_r - I_r\|_F^2 &= 0
\end{aligned}$$

This implies $\Theta_r^T\Theta_r = I_r$. Since $\Theta_r^T\Theta_r = \Theta_r\Theta_r^T = I_r$ this implies that Θ_r is a real-valued unitary matrix in the r-dimensional Euclidean space.

Proof of Theorem 4. *Given a rank-r matrix $XV_* \in \mathbb{R}^{m\times r}$, such that $V_*V_*^T = V_rV_r^T$ where V_r is a matrix with column vectors as the top-r right singular vectors of X, and a real-valued unitary matrix $\Theta_r \in \mathbb{R}^{r\times r}$ then $(XV_*\Theta_r)^T(XV_*\Theta_r)$ is a diagonal matrix Σ_r^2 where $\Sigma_r^2 = \operatorname{diag}(\sigma_1^2, \sigma_2^2, \cdots, \sigma_r^2)$ and σ_is are the top-r singular values of X if and only if $V_*\Theta_r = V_r$.*

$$\begin{aligned}
(XV_*\Theta_r)^T(XV_*\Theta_r) &= \Sigma_r^2 \\
\Theta_r^TV_*^T(X^TX)V_*\Theta_r &= \Sigma_r^2 \\
V_*^TX^TXV_* &= \Theta_r\Sigma_r^2\Theta_r^T \\
V_*V_*^TX^TXV_*V_*^T &= V_*\Theta_r\Sigma_r^2\Theta_r^TV_*^T
\end{aligned}$$

Using $V_*V_*^T = V_rV_r^T$ from **Theorem 2**,

$$\begin{aligned}
V_rV_r^TX^TXV_rV_r^T &= V_*\Theta_r\Sigma_r^2\Theta_r^TV_*^T \\
V_r\Sigma_r^2V_r^T &= (V_*\Theta_r)\Sigma_r^2(V_*\Theta_r)^T \\
(V_r - V_*\Theta_r)\Sigma_r^2(V_r - V_*\Theta_r)^T &= 0 \\
\|(V_r - V_*\Theta_r)\Sigma_r^2(V_r - V_*\Theta_r)^T\|_F &= 0
\end{aligned}$$

Using Frobenius norm to bound the matrix product,

$$\|\Sigma_r^2\|_F\|V_r - V_*\Theta_r\|_F^2 \geq 0$$

Since $\|\Sigma_r^2\|_F > 0$, equality is achieved if and only if $V_*\Theta_r = V_r$.

Remark. *Note that Θ_r is a rank-r unitary matrix wherein both rotation ($det(\Theta_r) = 1$) and reflection ($det(\Theta_r) = -1$) are valid since the order of the orthonormal vectors in the matrix $V_*\Theta_r = V_r$ do not alter $\|V_r - V_*\Theta_r\|_F$. In practice, Θ_r manifests itself predominantly as a rotation matrix during the iterative minimization using gradient descent.*

C ENERGY MINIMIZATION, LOSS SURFACE GEOMETRY, AND CONVERGENCE

In this section, we consider the energy minimization problem that constructs the projection space spanning the rank-r sub-space of a given data matrix. For ease of visualization, we consider a 2×2 matrix $X = \text{diag}(5, 1)$ with singular values 5 and 1 corresponding to right singular vectors $v_1 = [1, 0]^T$ and $v_2 = [0, 1]^T$, respectively. Our objective here is to extract a rank $r = 1$ approximation of this rank $f = 2$ matrix X. Certainly, this corresponds to identifying the right singular vector v_1 with singular value 5. The tail-energy surface (log-scale) corresponding to the $\|X\tilde{v}\tilde{v}^T - X\|_F$ is shown in **Fig. 10**. Here, $\tilde{v}$ is the test vector for a rank 1 approximation of X. The tail-energy is a bi-quadratic function in $\tilde{v}$ with 1 maximum, 2^r minima and 2^r saddle points, where r is the desired low-rank approximation of a given data matrix. Furthermore, all minima have the same tail energy: a property of bi-quadratic functions. For the current specific example, the two equal tail-energy minima correspond to v_1 and $-v_1$, respectively.

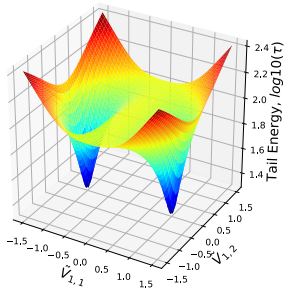


Figure 10: Surface plot for the bi-quadratic tail energy (log-scale) with 2 minima, 2 saddle points, and one maxima for $X = \text{diag}(5, 1)$.

Although the minimization problem is non-convex, convergence is guaranteed since any perturbed-gradient descent approach converges to either of the stable fixed points (minima). In other words, any test vector $\tilde{v}$ other than v_1 or $-v_1$ will increase the tail-energy and hence will not be the solution. The same argument applies for a high-dimensional dataset X where a low rank-r approximation is desired with the number of equal tail-energy minima corresponding to 2^r for all possible negative and positive combinations of the r right singular vectors $v_{i=1,\cdots,r}$. In effect, the stage 1 minimization problem constructs a right projection space $\tilde{V} = span\{v_1, \ldots, v_r\}$ that spans the top rank-r subspace of a given dataset X. A similar line of argument then applies to our stage 2 minimization problem as well. A mild limitation, that will be addressed in our future work, occurs when $X = \text{diag}(5, 5 + \epsilon)$, $0 \leq \epsilon << 1$, wherein the two right singular values cannot be resolved accurately (still better than Randomized SVD methods) without further considerations. This latter case, with near algebraic multiplicity in singular values is a special case for conventional SVD as well.

D NETWORK INTERPRETABILITY

As described before in Fig. 2, our network weights and outputs are strictly defined and incorporated as losses in the network minimization problem. We therefore refer to the problem informed (SVD) restrictions on the network weights as representation driven losses. This is in contrast to kernel regularization loss often considered to impose a weak requirement on the network weights to be small. The representation driven, orthonormality loss term, in Stage 1 enforces that the weights $\tilde{V}$ must be orthonormal or $(V_*^T V_* = I_r)$ for a desired rank-r which is greater than the rank-f of the data matrix X. We numerically verify the interpretability of the layer outputs and weights by considering two networks: (1) with, and (2) without the aforementioned orthonomality loss. For each of these two cases, two synthetic datasets are considered corresponding to $r \leq f$ and $r > f$. Please note that in a practical scenario f is an unknown and can be determined only by performing a full SVD of X. Therefore, numerically testing this aspect for our solver is necessary.

For the first case, we consider a synthetic data matrix $X_{15\times15}$ where the top 5 singular values are positive ($f = 5$) while the rest are zero. The objective is to extract the top 10 ($r = 10$) singular vectors where the desired rank is higher the the rank of the system itself. A total of four training runs

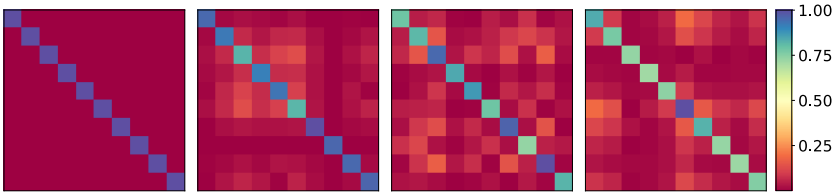


Figure 11: Synthetic Low Rank Case: Correlation Map of extracted vectors V_* over four runs. Orthonormality is imposed for the first run resulting in a diagonal structure. Absence of this condition results in a scatter.

are considered: one run for a network with the orthonormality condition imposed and three runs for another network without this additional constraints. **Fig. 11** shows the correlation map between the recovered vectors V_* for each of the four runs. Notice that only when the orthonormality criteria is not imposed, we get scatter away from the diagonal matrix, although all four runs converged to the same tail energy. Since the true rank of X is 5, the null space of X is of dimension 10. The absence of this orthonormality imposing, representation loss results in non-orthonormal vectors V_* spanning the low-rank range space.

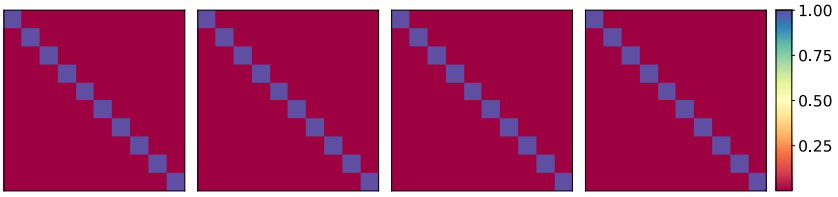


Figure 12: Synthetic Full Rank Case: Correlation Map of extracted vectors $\tilde{V}$ over four runs. Orthonormality loss now does not contribute and therefore all runs have a diagonal structure.

For the second case, we consider a full-rank, synthetic data matrix $X_{15\times 15}$ with $f = 15$. As before, we extract the top 10 singular vectors ($r = 10$) using four training runs: one with and three without imposing the orthonormality loss. Since the desired rank 10 system now itself is full rank, this additional loss term does not contribute, as desired. **Fig. 12** shows that the extracted vectors V_* remain orthonormal, for all the four runs, so as to minimize tail energy ($\|X(\tilde{V}\tilde{V}^T - I)\|_F$), as described in **Section 3.1** above. In fact, for any rank r approximation of a rank f system such that $r \leq f$, an arbitrary non-orthonormal matrix V_* will increase the tail energy and hence will not be a fixed point (solution) of our network minimization problem.

E Implementation Details

E.1 Choice of Activation Functions

All the activation functions in both stages are *Linear* with no biases. One might argue that this choice is not a neural approach, since all the activations are linear. However, please note that singular vectors are *linearly separable orthogonal features* of a dataset, and therefore any other choice of activation function will result in approximation errors. Since the elements of singular vectors are in $[-1, 1]$, a choice of *relu* activation is problematic. A simple verification is to approximate a straight line segment in $[-1, 1]$ with *tanh* activation, only to realize that the approximation error $\to 0$ as the number of neurons $\to \infty$. These arguments can also be verified by replacing linear activation in Stage 1 by any non-linear activation only to find that the tail energy bound cannot be satisfied. Note that given a small matrix, one can calculate the right singular vectors and substitute them directly as our network weights to confirm this tail energy bound.

E.2 Training, Validation, and Testing Split

An issue with training and validation split in matrix decomposition problems is that the error norm cannot be bounded in a deterministic manner or computationally verified. For Singular Value Decomposition of a given data matrix X differs from SVD on a truncated dataset $\hat{X}$ in it's singular triplets (singular values and vectors). Ensuring these triplets do not change over an arbitrary split is a non-trivial computational task.

Remark. *For dataset X, an arbitrary training/validation split results in a varying dataset $\hat{X}$ wherein the norm $\|X_{pred} - \hat{X}\|_F$ changes according to the split. Since the desired features are unknown a priori, a consistent truncated dataset $\hat{X}_c$ that spans the same space as the full data X cannot be obtained using an arbitrary split.*

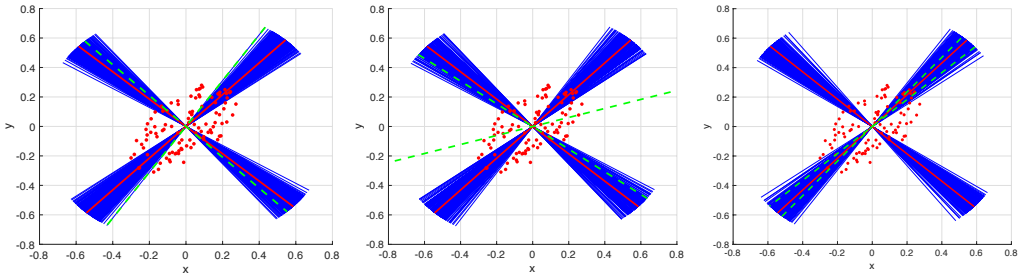


Figure 13: Variation in extracted right singular vectors (dashed green lines) for three runs with each run comprising of 200 different train/test data splits with an $80/20$ ratio. The solid red lines indicate the ground truth singular vectors when using the entire unsplit dataset (red dots). Extracted singular vectors deviate increasingly from left to right over the three runs.

This results in a large variance in extracted features over multiple training/testing splits since the span of $\hat{X}$ itself is changing with each split. **Fig. 13** shows a numerical experiment with a synthetic (rank-2) dataset containing 100 samples drawn (red dots) from a 2D ellipse with major and minor axes 2 and 1, respectively. The ground truth right singular vectors over the unsplit dataset are computed using conventional SVD (solid red lines). Next the dataset was split with an $80/20$ ratio as the training/testing split for 200 different realizations per run. The mean/expectation of the extracted right singular vectors (dashed green line) over the 200 realizations are then reported. One can easily see that the expected right singular vectors vary quite substantially compared to the ground truth over the three runs indicating that an arbitrary split of the dataset does not ensure accuracy.

E.3 Data Streaming

Given a data matrix $X^{m\times n}$, we stream the data along the smaller dimension assuming the user prescribed rank-r is such that $r \leq min(m, n)$. For the sake of simplicity we assume that the data matrix has m samples and n features, where $m > n$, and consequently feature vectors of samples are streamed in batches. We rely upon the built-in Keras **fit_generator** class for data streaming from the secondary memory (HDD). For a big data matrix X that cannot be loaded into the main memory, this allows us to mimic the modality of data residing on an external server. Thus, given a pointer to the data, the function yields a batch of specified size for the network to train on for specific epochs. This ability saves main memory load and allows us to process bigger datasets on smaller main-memory machines than reported in prior works.

Note that for the stage-1 network to converge to a desired tolerance, we require multiple passes (empirically ≤ 5) over the original data streamed batchwise. Therefore for Stage 1, the input data is never persistently present in the main memory of the remote machine. The output data is dumped onto the secondary memory assuming that storing a low rank approximation is still main memory intensive. For Stage 2, this low rank approximation in the secondary memory is streamed as input, and the extracted singular values and vectors are saved in main memory.

E.4 Setup and Training

All experiments were done on a setup with Nvidia 2060 RTX Super 8GB GPU, Intel Core i7-9700F 3.0GHz 8-core CPU and 16GB DDR4 memory. We use Keras (Chollet, 2015) running on a Tensorflow 2.0 backend with Python 3.7 to train the networks presented in this paper. For optimization, we use AdaMax (Kingma & Ba, 2014) with parameters (lr= 0.001) and 1000 steps per epoch.

E.5 Error Metrics

As discussed before in **Section 2.1**, since relative errors in tail energies do not imply similar errors at scales in extracted singular factors, we rely upon additional error metrics on the extractor factors for

performance comparison and benchmarking. In the following X and $\hat{X}$ are used to denote the true and the reconstructed data matrices.

- **Scree Error:** Absolute difference between true and approximated singular values.

$$scree_{err}(r) = |\sigma_i(X) - \hat{\sigma}_i(\hat{X})| \quad \forall\, i \in [1, r]$$

- **Reconstruction Error:** Frobenius norm error of the true data and its rank-r approximation.

$$frob_{err}(r) = \|X - \hat{X}\|_F^2 - \|X - X_r\|_F^2$$

- **Spectral Error**: 2-norm of the singular value of the true data and its rank-r approximation.

$$spectral_{err}(r) = \|X - \hat{X}\|_2 - \|X - X_r\|_2$$

- **Chi Square Statistic**: Deviation between the true and approximated singular vectors.

$$\chi_e^2 rr(r) = 1 - \frac{1}{r}\|Corr(v_{[:r]}(X), \hat{v}_{[:r]}(\hat{X}))\|_F$$

Here, σ_is are the true singular values and X_r is the desired rank-r approximation of X using conventional SVD as the baseline for benchmarking. Under perfect recovery, all error metrics are expected to approximately achieve zero at machine precision. All of our numerical experiments were performed on a GPU using single (32-bit) precision floating point operations. Therefore, the tail energies are expected to be correct to up to 8 significant digits in all the subsequent calculations. In the following sections, $\hat{X}$ is replaced by approximations from SketchySVD and Range-Net.

E.6 Loss Profile

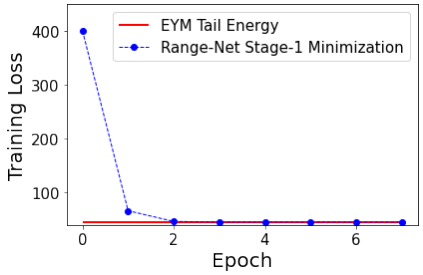


Figure 14: Range-Net Stage-1 training loss for Parrot image. The network minimization loss converges to the EYM Tail Energy bound within five epochs and stabilizes for further epochs.

F Additional Experiments

F.1 Feature Extraction: Graph (Eigen)

Table 5: Description and Metrics of the Network Graphs for Range-Net

Dataset	Nodes	Edges	rank	err_{fr}	err_{sp}	χ^2_{err}
Airlines (air)	235	2101	200	0	0	0.011
Twitter (air)	3556	188712	200	0	0	0.014
Wikivote (Leskovec & Krevl, 2014)	8297	103689	200	0	0	0.027
Wikipedia (lev)	49728	941425	100	4.27e-6	1.23e-7	0.034
Slashdot (Leskovec & Krevl, 2014)	82168	948464	100	8.56e-6	6.92e-7	0.045

Large scale networks occur in many applications where SVD is primarily used to identify the most important nodes or as a pre-processing step for community detection. For these kind of graph based datasets, we either perform SVD or Eigen decomposition on the graph, depending on the format in which the data arrives. We demonstrate results on the following graphs of varying size, tabulated in **Table 5**. If the data arrives directly in the form of an adjacency matrix, we can perform SVD or Eigen decomposition on it directly. For cases, where an adjacency list is provided, a pre-processing step is required to convert the list representation in a sparse vector.

Since an Eigen decomposition problem is a special case of SVD, where the data matrix is symmetric positive semi-definite, Range-Net is directly applicable. The benchmark was generated for smaller graphs using a conventional SVD solver. For larger graphs, a similar benchmark was constructed using the *irlba* routine by Baglama & Reichel (2005). **Table. 5** shows the error metrics for all the graphs, where consistently low values are observed for Frobenius and Spectral error metrics.

F.2 Navier-Stokes Simulated Data (SVD)

For our next numerical experiment, we rely upon synthetic data generated using a Navier Stokes flow simulator for an incompressible fluid. The flow data is available on tensor-product grid on two-spatial and one temporal dimensions of size $(w, h, t) = (100 \times 50 \times 200)$. For each point on the grid, velocity vector values are available in both x and y spatial dimensions for 200 time instances. The fully-developed, flow pattern exhibits a periodicity in the time dimension at approximately every ~ 60 time step that can be identified using SVD as characteristic modes. The data is therefore reshaped into a spatial vector for each time instance resulting in a spatio-temporal matrix $X \in \mathbb{R}^{5000\times 200}$. For comparison purposes, we use only the x-direction stream velocity.

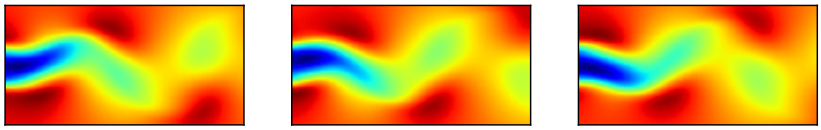
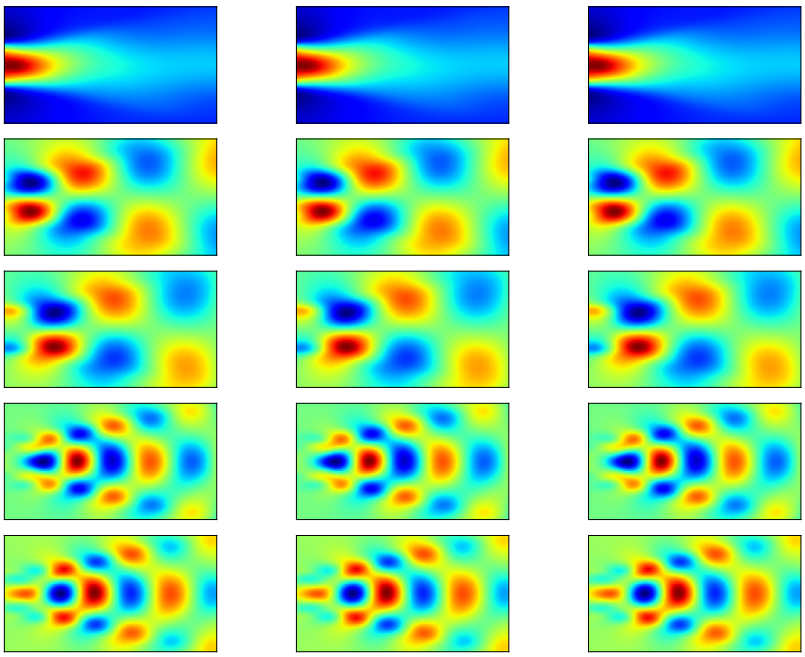


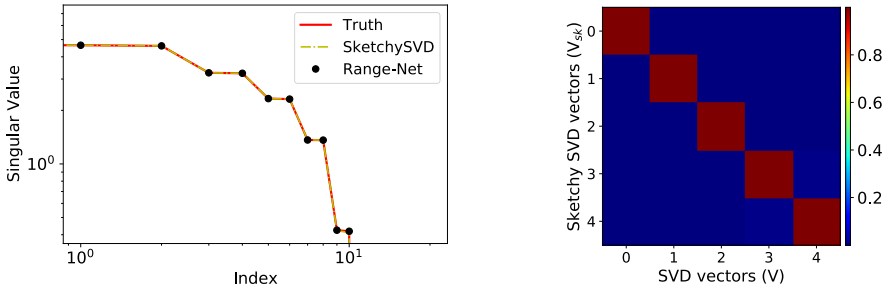
Figure 15: From left to right: x-direction stream velocity at times t = 0, 100, and 200



(a) SketchySVD (b) Truth (c) Range-Net

Figure 16: Reshaped U indicative of dynamic modes, corresponding to top six right singular vectors for $r = 5$.

Fig. 15 shows the evolution of the stream velocity over three time instances and the inherent time-periodic nature of the data. Notice the central-left region of primary flow across all three images. **Fig. 16** shows the reshaped U vectors for SketchySVD, conventional SVD and Range-Net. Notice that the images corresponding to the first left singular mode (also called dynamic modes) captures a notion of the primary flow in the left-center part. The second one captures spatial variations of the flow as time progresses. For all the three methods, all the modes have similar solution visually. Note that



(a) Singular Values (b) Correlation

Figure 17: SketchySVD and Range-Net (a) extracted singular values and (b) cross correlation between estimated and true (conventional SVD) right singular vectors for $r = 5$ on the Navier-Stokes data.

Range-Net computes a rank $r = 5$ approximation without oversampling, wherein SketchySVD relies upon memory intensive sketchy projections of ranks $k = 21, s = 43$ to arrive at the solution. Overall, for this low-rank dataset both SketchySVD and Range-Net perform reasonably well qualitatively looking at the features in **Fig.** 16 and the singular value spectrum in **Fig.** 17, because this synthetic dataset is extemely low-rank ($f = 10$).

Remark. *For a given data matrix $X \in \mathbb{R}^{m\times n}$ of rank-f ($f \leq \min(m, n)$) SketchySVD generates low error approximations if the oversampled rank k is such that $k \geq f$. For full rank tall skinny matrices, this implies that $k \geq \min(m, n)$. For cases where $k < f$ (full rank or otherwise), SketchySVD accrues large approximation errors resulting in incorrect SVD factors.*

From a use-case point of view, randomized SVD generates low-error factors for full-rank, tall skinny matrices ($X \in \mathbb{R}^{m\times n}$) only when the oversampled rank $k \geq n$. This poses a serious limitation for all applications where this requirement is not met and consequently randomized SVD algorithms accrue large approximation errors in SVD factors as shown in the Sandy Big Data case study below and **Appendix** F.4. Note that given a big data matrix X determining the rank f of X is unknown and therefore selecting an oversampled rank k such that $k \geq f$ is impractical in such cases.

Remark. *Once the SVD factors are extracted, SketchySVD cannot be independently verified without performing a full SVD. In contrast, Range-Net is independently verifiable since Stage-1 of Range-Net cannot return orthonormal vectors if the vectors do not span the rank-r subspace of a given data X.*

F.3 High Rank Approximation: Sandy Big Data

This section provides an addendum to the numerical results presented in **Section** 4.3 of the main text. **Fig.** 18 shows evolution of Hurricane Sandy for two time instances. Similar to the Navier-Stokes simulation data, **Fig.** 19 shows three dynamic modes corresponding to rank $1, 20, 50, 100$ singular values. As shown, our results are in good agreement with conventional SVD whereas, Sketchy SVD shows substantial deviations after the first 50 dynamic modes. **Fig.** 20 shows the scree-error in the singular values extracted by SketchySVD and Range-Net.

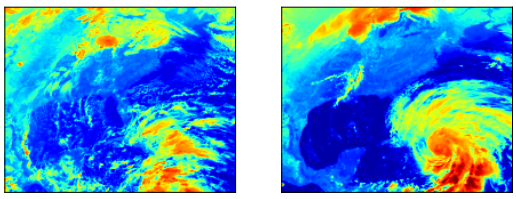
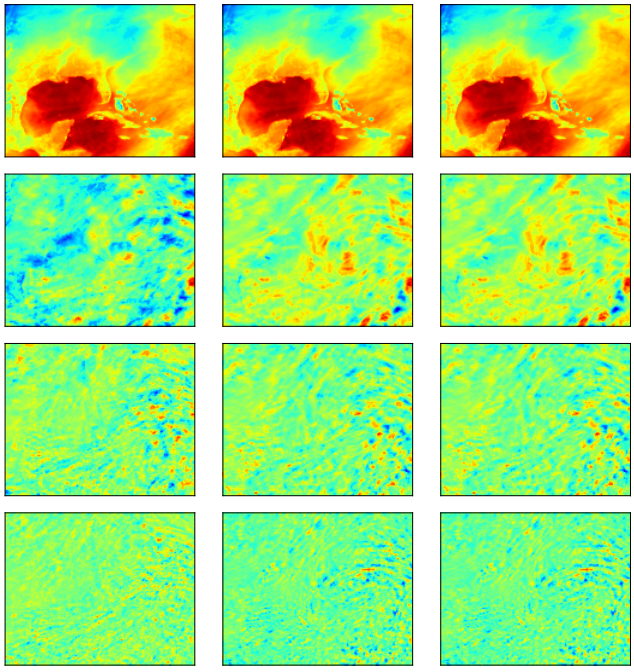


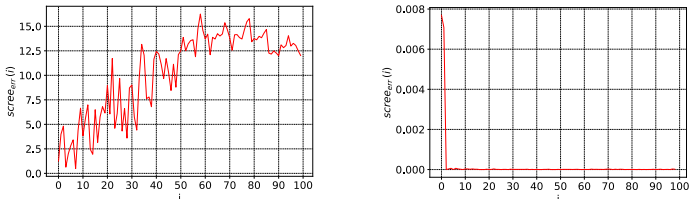
Figure 18: Satellite image captures of hurricane Sandy over the Atlantic ocean at $t = 0$ (left) and $t = 200$ minutes approximately (right).

Fig. 21 shows the cross correlation of the right singular vectors and scree-error in the corresponding singular values extracted by SketchySVD and Range-Net. Note that for a rank-100 approximation, SketchySVD extracted right singular vectors start deviating after rank-10 as shown in **Figs.** 21 while the singular values deviate quite substantially from rank-1. Range-Net on the other hand is in excellent agreement with the right singular vectors and values for all desired 100 indices.



(a) SketchySVD (b) Truth (c) Range-Net

Figure 19: Reshaped U_i indicative of dynamic modes, corresponding to $i = 1, 20, 50, 100$ for $r = 100$ (oversampled rank $k = 401$ for SketchySVD. The dynamic mode approximation error stand out visually for SketchySVD for indices $20, 50, 100$. Our method does not have such artifacts.



(a) SketchySVD (b) Range-Net

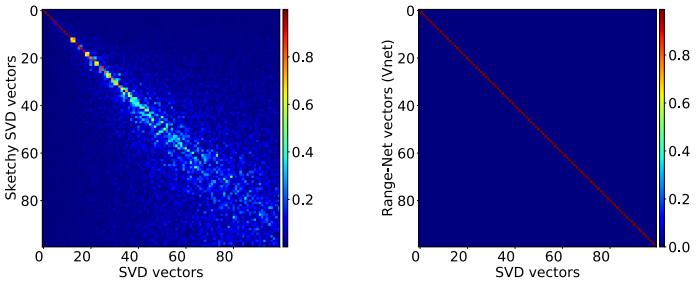
Figure 20: Scree-error in singular values for (a) SketchySVD and (b) Range-Net where a conventional SVD is used as the baseline in scree-error metric.

We point out that accuracy is a matter of special concern in scientific computations. Any compression that results in substantial loss of information or obscuring an otherwise identifiable feature in recorded observations directly culls our capacity to make scientific improvements. Consequently, any exploratory data analysis, however big or small, must accurately identify the underlying features. Range-Net achieves the lower bound on tail-energy given by EYM theorem to ensure an accurate resolution in big data setting. Note that increasing the sensor resolution implies that we are interested in exploring and understanding the high-frequency features (lower singular values) of the data.

F.4 Low Rank Approximation: Sandy Big Data

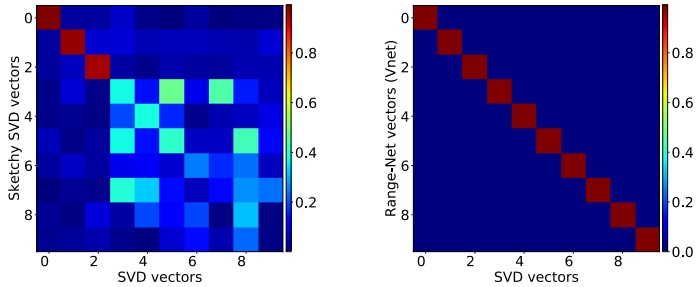
In this experiment, we extract the rank-10 SVD factors using SketchySVD and Range-Net for the Sandy dataset. The oversampled ranks for sketchy SVD are $k = 4r + 1 = 41$ and $s = 2k + 1 = 83$ where $k, s \ll \min(m, n)$.

As before, **Fig. 22** show the cross-correlation between the extracted and true (conventional SVD) right singular vectors using SketchySVD and Range-Net. **Fig. 23** shows the scree error in the extracted singular values for the two methods with singular values from conventional SVD as the baseline. Finally, **Fig. 24** shows a comparison between extracted dynamic modes corresponding to indices $i = [1, 4, 7, 10]$ from SketchySVD, conventional SVD, and Range-Net. One can easily see



(a) SketchySVD (b) Range-Net

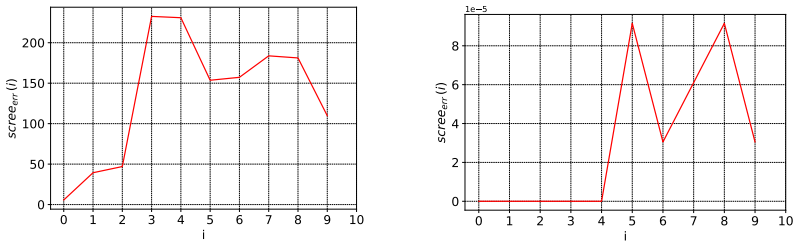
Figure 21: Cross-correlation between extracted and true (conventional SVD) right singular vectors for (a) SketchySVD and (b) Range-Net for a rank $r = 100$ approximation. SketchySVD deviates substantially after index 10 (although sketching at sizes $k = 401$ and $s = 803$) while Range-Net is in good agreement for all the 100 indices.



(a) SketchySVD (b) Range-Net

Figure 22: Cross-correlation between extracted and true (conventional SVD) right singular vectors for (a) SketchySVD and (b) Range-Net for a rank $r = 10$ approximation. SketchySVD deviates substantially after index 3 while Range-Net is in good agreement for all the 10 indices.

that Sketchy SVD extracted dynamic modes/right singular vectors deviate quite substantially for $i = [4, 7, 10]$.

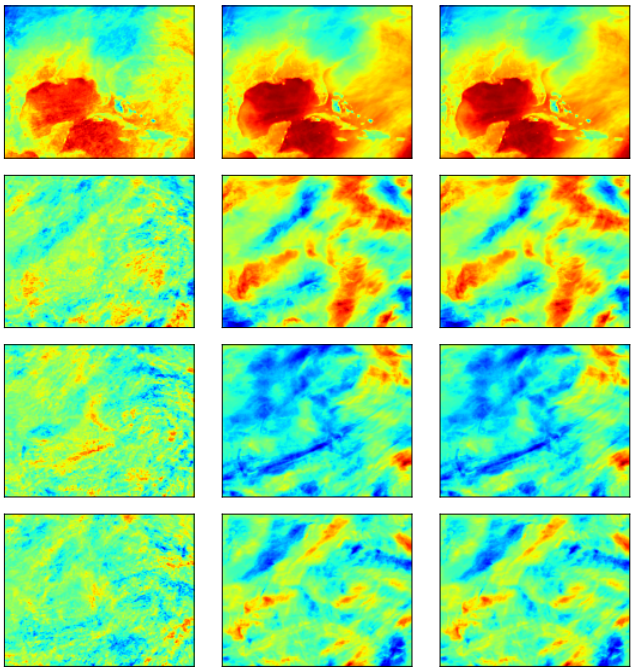


(a) SketchySVD (b) Range-Net

Figure 23: Scree-error in singular values for (a) SketchySVD and (b) Range-Net where a conventional SVD is used as the baseline in scree-error metric. Note that for Range-Net the error is at a scale of 10^{-5}, 7 orders of magnitude apart from SketchySVD (10^2).

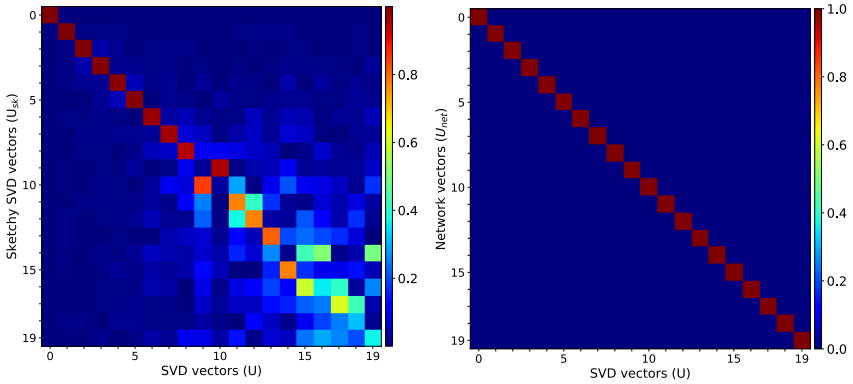
F.5 PARROT: ADDENDUM CORRELATION OF LEFT SINGULAR VECTORS

In the following, we show the deviation of left singular vectors, computed using Range-Net and Sketchy SVD, from the conventional SVD computed left singular vectors as a baseline. As shown before, random sketching introduces irreducible errors in randomized SVD methods resulting in this unchecked deviation. **Fig. 25** shows a comparison of the cross-correlation against the common baseline for both Range-Net and Sketchy SVD.



(a) SketchySVD (b) Truth (c) Range-Net

Figure 24: Reshaped U_i indicative of dynamic modes, corresponding to $i = 1, 4, 7, 10$ for $r = 10$. The dynamic mode approximation error stand out visually for SketchySVD for indices $1, 4, 7, 10$. Our method does not have such artifacts.



(a) SketchySVD (b) Range-Net

Figure 25: Cross-correlation between true (conventional SVD) and extracted left singular vectors from (a) SketchySVD (b) Range-Net for a rank-$r = 20$ approximation of the Parrot image.

G SKETCHY SVD IMPLEMENTATION

A brief outline of the single-pass Sketch-SVD algorithm from Tropp et al. (2019). Note that for the numbers reported in terms of storage, we implemented this code with additional memory optimization and sparse matrices.

Algorithm 1 Sketchy SVD

Input: $X \in \mathbb{R}^{m \times n}$, r : expected rank
Output: $\tilde{X} \in \mathbb{R}^{m \times k}$ the approximated rank k-dim data

Initialize $k = 4r + 1, s = 2k + 1$ ▷ Oversampling parameters
Projection maps: $\Upsilon \in \mathbb{R}^{k \times m}, \Omega \in \mathbb{R}^{k \times n}, \Phi \in \mathbb{R}^{s \times m}, \Psi \in \mathbb{R}^{s \times n}$
Projection matrices: $A \in \mathbb{R}^{k \times n}, B \in \mathbb{R}^{m \times k}, Z \in \mathbb{R}^{s \times s}$ as empty
for $i = 1 : n$ **do** ▷ Streaming Update
    Form $H \in \mathbb{R}^{m \times n}$ as a sparse empty matrix
    $H(i, :) = X(i, :)$ ▷ Streamed columns
    $A \leftarrow A + \Upsilon H$ ▷ Update Co-Range
    $B \leftarrow B + H\Omega^T$ ▷ Update Range
    $Z \leftarrow Z + \Phi H \Psi^T$ ▷ Update Core Sketch
$Q \in \mathbb{R}^{m \times k} \leftarrow qr\_econ(B)$ ▷ Basis for Range
$P \in \mathbb{R}^{n \times k} \leftarrow qr\_econ(A^T)$ ▷ Basis for Co-Range
$C \in \mathbb{R}^{s \times s} \leftarrow ((\Phi Q) \setminus Z)/(\Psi P)$ ▷ Core Matrix
$[U, \Sigma, V^T] \leftarrow svd(C)$ ▷ Full SVD of Core Matrix
$\Sigma \in \mathbb{R}^{r \times r} \leftarrow \Sigma[1 : r, 1 : r]$ ▷ Pick top r
$U \in \mathbb{R}^{k \times r} \leftarrow U[:, 1 : r]$ ▷ Pick top r
$V^T \in \mathbb{R}^{r \times k} \leftarrow V^T[1 : r, :]$ ▷ Pick top r
$U \in \mathbb{R}^{m \times r} \leftarrow QU$ ▷ Project to Row Space
$V^T \in \mathbb{R}^{r \times n} \leftarrow PV^T$ ▷ Project to Column Space
$\tilde{X} \in \mathbb{R}^{m \times n} \leftarrow U\Sigma V^T$ ▷ Approximation