

ME EN 7960: Scientific Machine Learning Fall 2024

Midterm Project

Milena Belianovich
February 28, 2025

Chosen data (Problem 1): The chosen dataset was generated using the MOOSE framework and represents a 3D simulation of an object being pulled apart and rotated. This dataset, developed within Varun Shankar's lab (by Alberto Cattaneo), is used for testing different solver methods and features a rotating block with a cohesive zone. For our analysis, we extract a 2D slice with velocity vector data, comprising at least 20 time-steps, to perform dimensionality reduction and denoising.

1 Dimensionality Reduction on Spatiotemporal Data

Given a spatiotemporal dataset, our goal is to perform Principal Component Analysis (PCA) to identify dominant modes and analyze the underlying structure. The dataset consists of velocity fields in two components, v_x and v_y , across multiple time steps. PCA is performed using Singular Value Decomposition (SVD) to extract dominant patterns, quantify variance distribution, and investigate dimensionality reduction capabilities.

Solution. To illustrate the temporal evolution of the dataset, we provide an animation (Figure 1) that visualizes the movement of data points over time. This animation demonstrates the underlying structure and highlights the necessity of dimensionality reduction techniques (full animation is only viewable in Adobe Acrobat Reader).

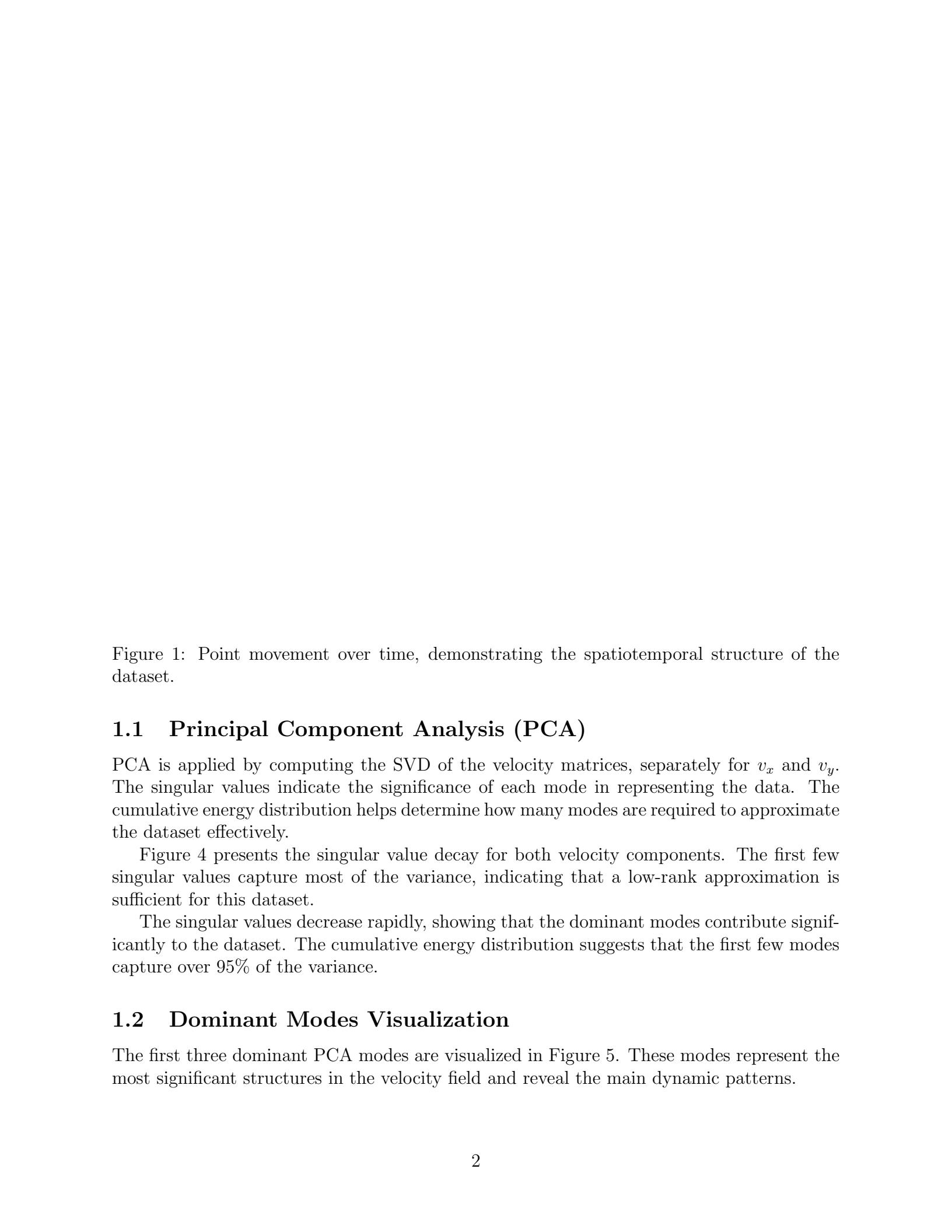


Figure 1: Point movement over time, demonstrating the spatiotemporal structure of the dataset.

1.1 Principal Component Analysis (PCA)

PCA is applied by computing the SVD of the velocity matrices, separately for v_x and v_y . The singular values indicate the significance of each mode in representing the data. The cumulative energy distribution helps determine how many modes are required to approximate the dataset effectively.

Figure 4 presents the singular value decay for both velocity components. The first few singular values capture most of the variance, indicating that a low-rank approximation is sufficient for this dataset.

The singular values decrease rapidly, showing that the dominant modes contribute significantly to the dataset. The cumulative energy distribution suggests that the first few modes capture over 95% of the variance.

1.2 Dominant Modes Visualization

The first three dominant PCA modes are visualized in Figure 5. These modes represent the most significant structures in the velocity field and reveal the main dynamic patterns.

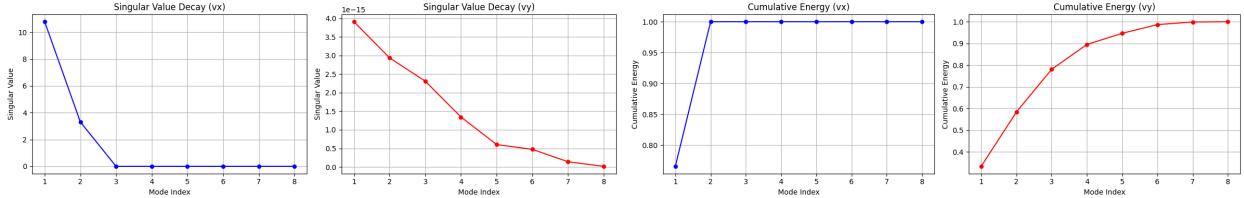


Figure 2: Singular value decay.

Figure 3: Cumulative energy distribution.

Figure 4: Singular value decomposition and cumulative energy for the velocity components.

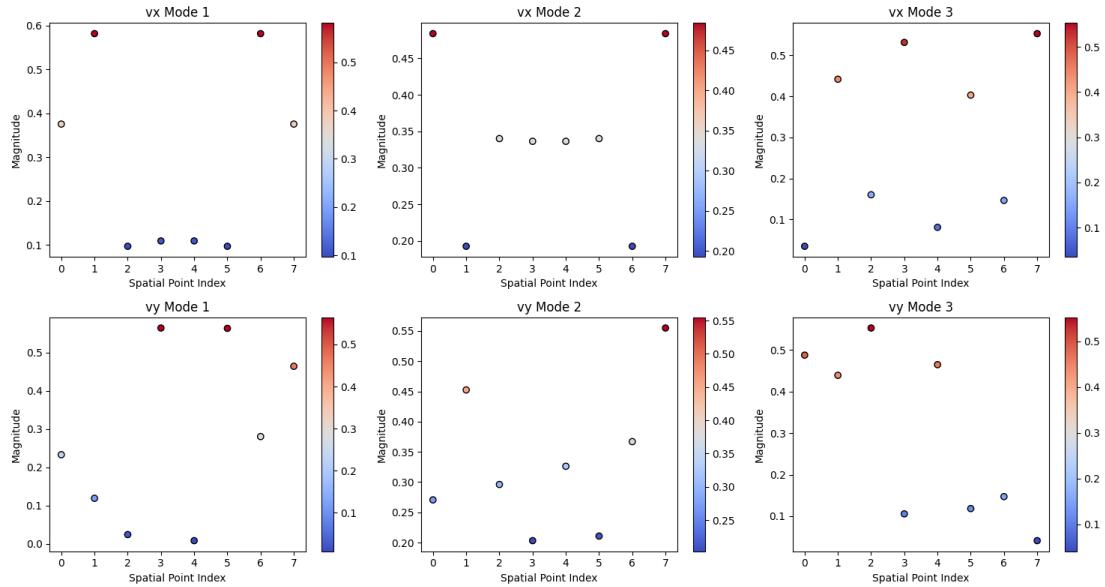


Figure 5: First three dominant PCA modes for v_x and v_y .

These modes indicate that most of the flow information is captured in the first few modes, while higher-order modes contribute minor details.

1.3 Corruption of Data

To evaluate the robustness of PCA under noise, we introduce two types of corruption to the dataset:

- 1. Random noise:** Noise is added to 80% of the dataset by perturbing each entry with a normally distributed random value. This simulates widespread measurement errors.
- 2. Localized noise:** Noise is introduced in a confined spatial region, mimicking distortions that may arise from localized defects or external disturbances.

The noise is generated from a Gaussian distribution with zero mean and unit standard deviation, scaled by a noise factor ζ , where $\zeta \in \{0.05, 0.1, 0.2\}$. The noisy dataset is computed as:

$$x_{\text{noisy}} = x + \zeta \cdot \text{rand} \cdot x, \quad (1)$$

where rand is a normally distributed random variable.

1.3.1 Effect on Singular Values and Cumulative Energy

The impact of noise on the singular values and cumulative energy is shown in Figures 9 and 13. For each noise level, we compare the singular value decay before and after corruption.

The introduction of random noise results in a distortion of the singular values, particularly in the lower-order modes. As the noise level increases, more singular values become significant, indicating that PCA struggles to distinguish signal from noise.

Similarly, the effect of localized noise is presented in Figure 13.

In contrast to random noise, localized noise tends to affect only a subset of the dominant modes, leading to localized degradation in the reconstructed data. The effect is less widespread but can still introduce significant distortions in specific regions of the dataset.

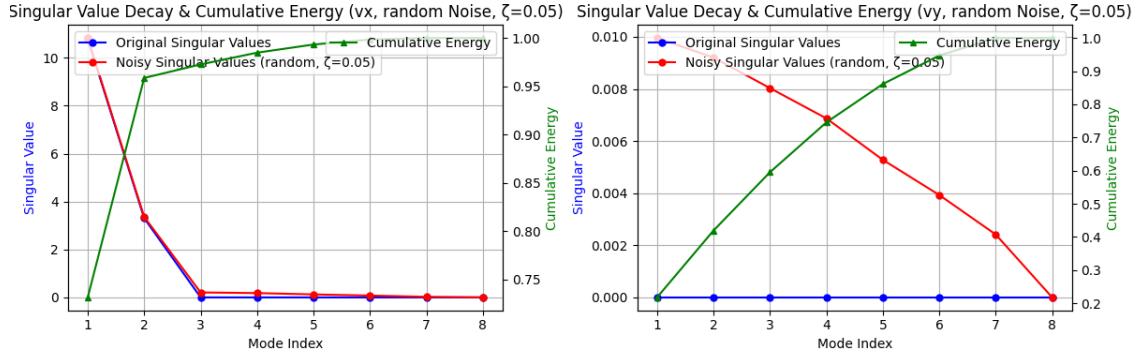


Figure 6: Random noise, $\zeta = 0.05$

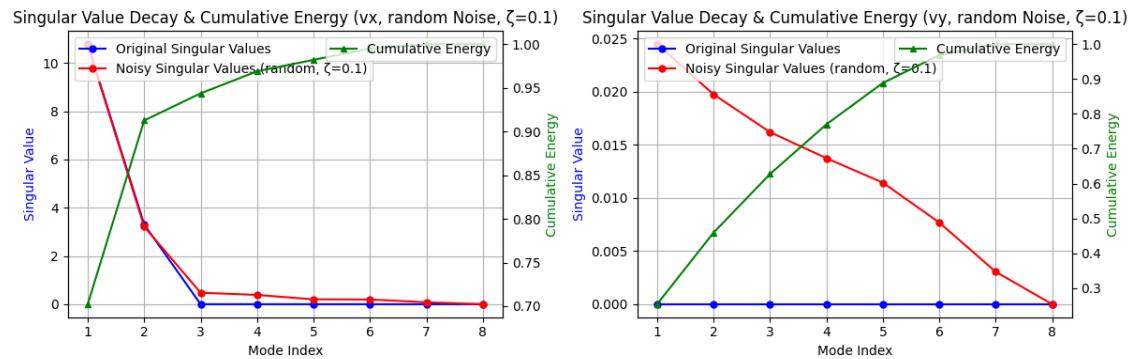


Figure 7: Random noise, $\zeta = 0.1$

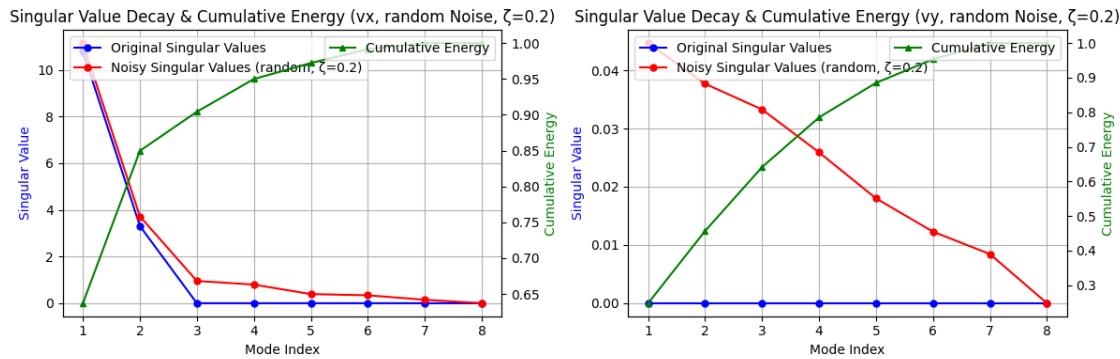


Figure 8: Random noise, $\zeta = 0.2$

Figure 9: Singular values and cumulative energy distribution under random noise.

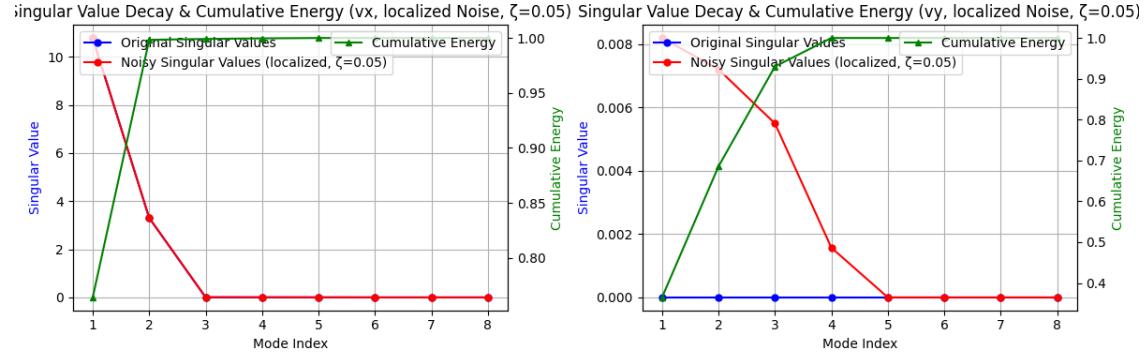


Figure 10: Localized noise, $\zeta = 0.05$

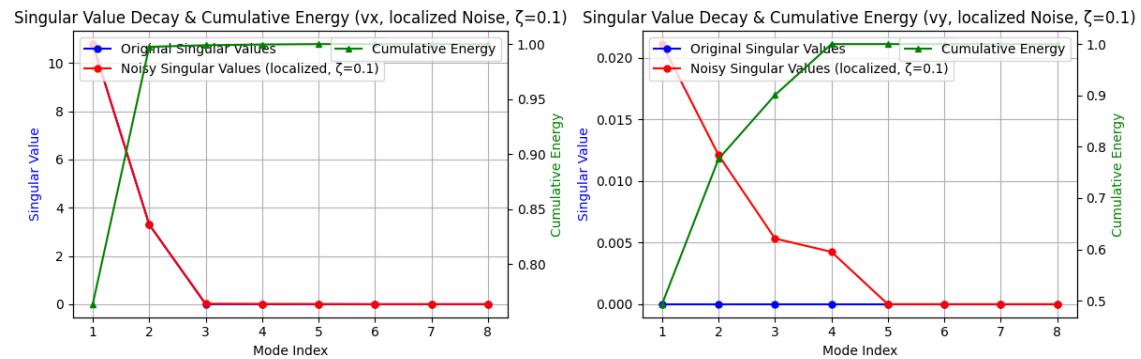


Figure 11: Localized noise, $\zeta = 0.1$

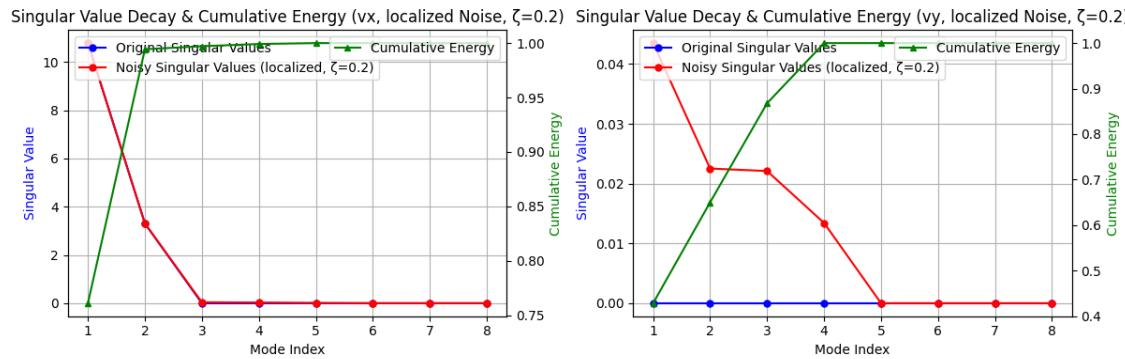


Figure 12: Localized noise, $\zeta = 0.2$

Figure 13: Singular values and cumulative energy distribution under localized noise.

1.3.2 Discussion

- **Impact of random noise:** The entire spectrum of singular values is affected, leading to a loss of clear separation between dominant and non-dominant modes. This suggests that PCA struggles with distinguishing signal from noise when corruption is widespread.
- **Impact of localized noise:** The degradation is more constrained to certain modes, meaning some of the dominant structures remain identifiable. However, the corrupted region's contribution to variance increases disproportionately.
- **Energy retention:** As noise levels increase, more modes become necessary to capture the dataset's variance, reducing the effectiveness of low-rank approximations.

These observations motivate the use of Robust PCA (RPCA) to mitigate noise-induced artifacts, as discussed in the next section.

1.4 Robust PCA (RPCA)

While PCA is effective for dimensionality reduction, it is not inherently robust to noise. Robust PCA (RPCA) is an extension that separates the low-rank structure from sparse noise, making it suitable for denoising. Here, we apply RPCA to the corrupted velocity data and analyze its effectiveness.

RPCA is performed by decomposing the data matrix V into two components:

$$V = L + S, \quad (2)$$

where L is a low-rank matrix capturing the clean underlying structure, and S is a sparse matrix representing noise and anomalies. The optimization problem is formulated as:

$$\min_{L,S} \|L\|_* + \lambda \|S\|_1 \quad \text{subject to} \quad V = L + S, \quad (3)$$

where $\|L\|_*$ denotes the nuclear norm (sum of singular values) promoting low rank, and $\|S\|_1$ represents the L_1 norm enforcing sparsity. The regularization parameter λ controls the trade-off between these terms, with the best choice determined experimentally.

1.4.1 RPCA Results

Figures 17 and 21 show the singular value decay and cumulative energy distribution for different noise levels after RPCA has been applied. These results are compared against the original corrupted data.

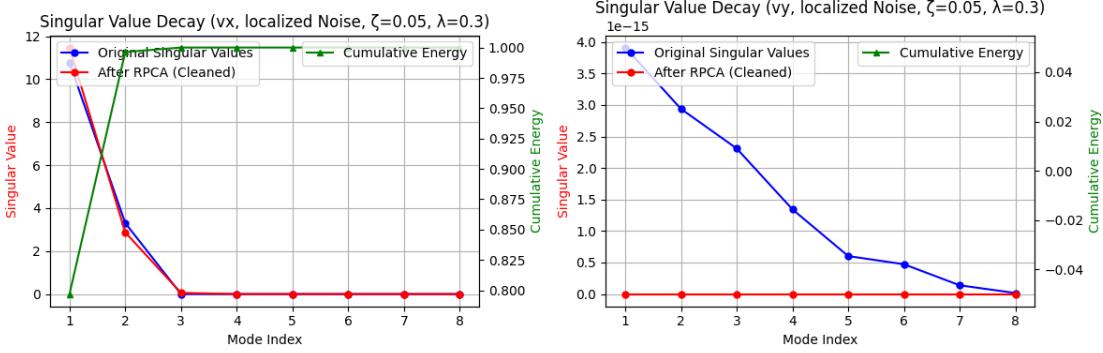


Figure 14: Localized noise, $\zeta = 0.05$

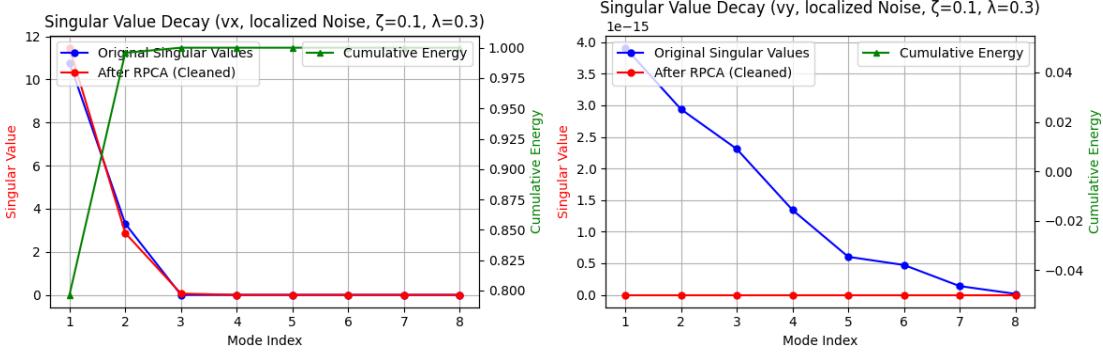


Figure 15: Localized noise, $\zeta = 0.1$

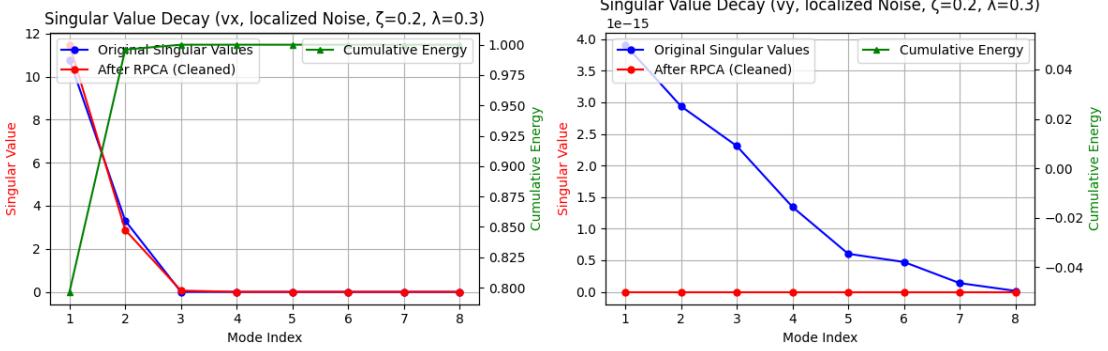


Figure 16: Localized noise, $\zeta = 0.2$

Figure 17: RPCA results for localized noise at different levels.

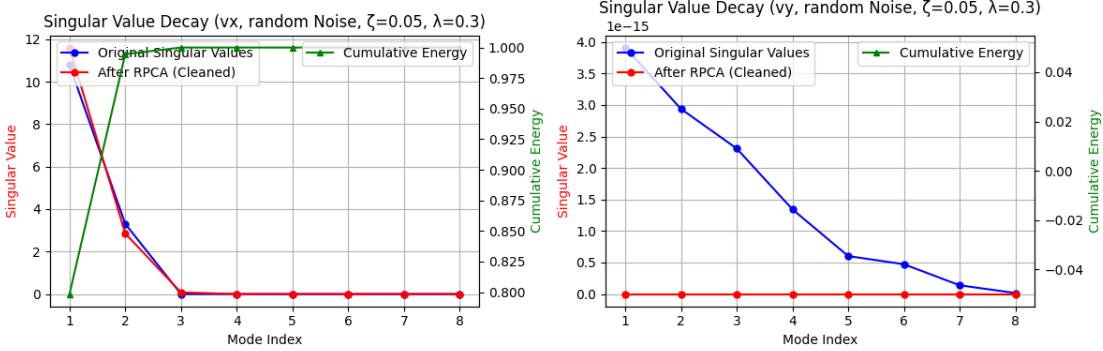


Figure 18: Random noise, $\zeta = 0.05$

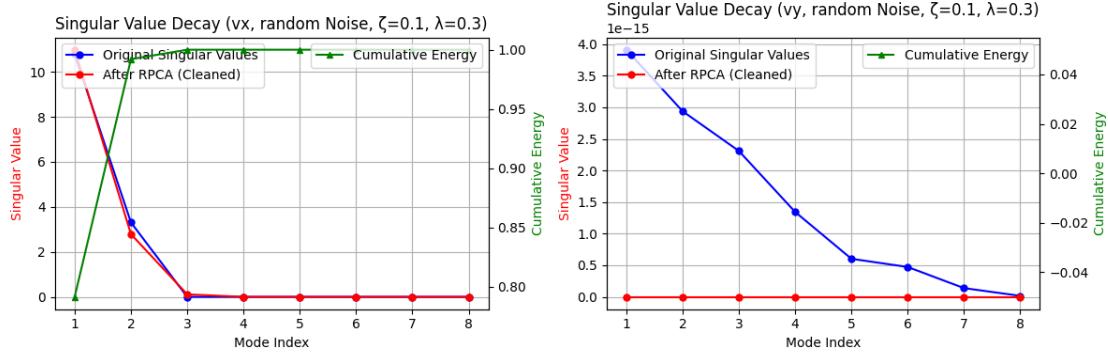


Figure 19: Random noise, $\zeta = 0.1$

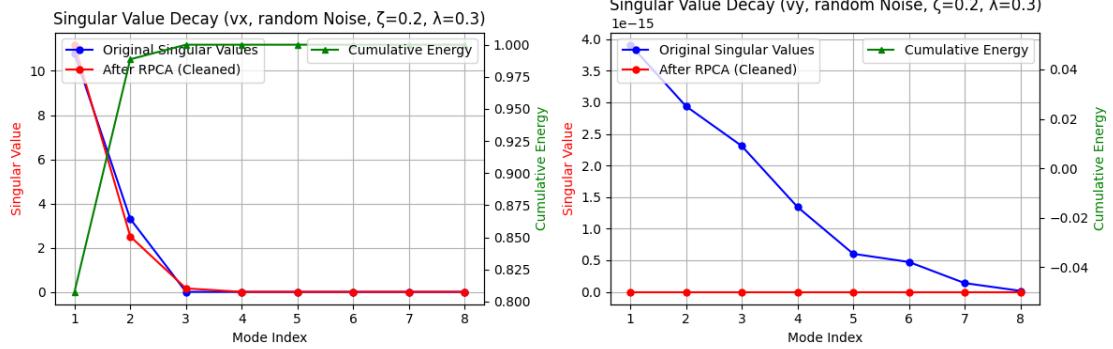


Figure 20: Random noise, $\zeta = 0.2$

Figure 21: RPCA results for random noise at different levels.

1.4.2 Comparison with Original Data

To evaluate RPCA's effectiveness, we compare the singular value spectrum before and after applying RPCA, as shown in Figure 22. The RPCA-cleaned data retains only the dominant modes while significantly reducing the influence of noise.

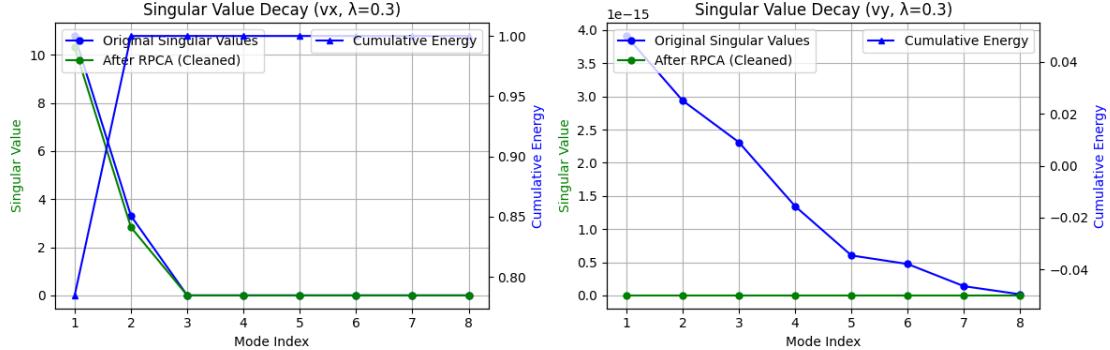


Figure 22: Comparison of singular value decay and cumulative energy before and after RPCA ($\lambda = 0.3$).

1.4.3 Discussion

The RPCA results indicate a significant improvement in preserving the dominant structures while eliminating noise. Key observations include:

- The RPCA-cleaned singular values align closely with the original, uncorrupted data.
- Higher noise levels result in stronger low-rank approximations, emphasizing RPCA's effectiveness.
- Random noise is better mitigated compared to localized noise, which may still introduce small artifacts.
- The cumulative energy distribution remains stable after RPCA, confirming that the denoising process retains essential structural information.

Overall, RPCA successfully extracts the underlying clean velocity field, making it a robust method for noise-affected spatiotemporal data.

2 Compressed Sensing and Fourier Reconstruction

Assume we have time-series data that we can represent as a Fourier series. We can define such a signal as:

$$u(t) = a_0 + \sum_{i=1}^N a_i \cos(i\omega t) + \sum_{i=1}^N b_i \sin(i\omega t). \quad (4)$$

Define such a signal with $N = 3$. You may arbitrary select the coefficients as well as the frequency ω . Generate the signal for $0 < t < T$ and ensure you have a high enough resolution. Select T such that your data contains a few of the periods. This is your ground-truth data that you can use for calculating your data reconstruction accuracy below.

Solution. The high-resolution ground truth signal is generated using 2000 time points, ensuring smooth representation. This signal serves as the reference for evaluating reconstruction accuracy.

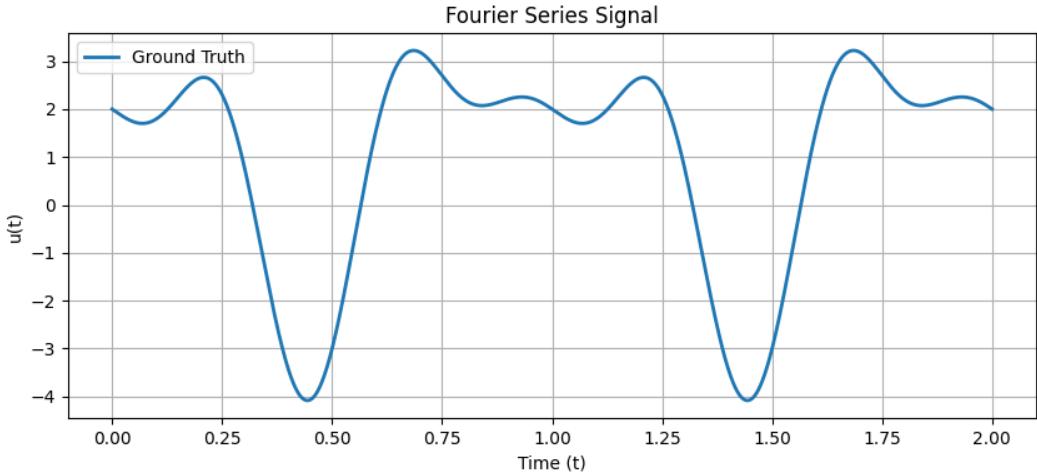


Figure 23: Ground-truth Fourier series signal with $N = 3$ and $\omega = 3$.

- Now imagine you are an experimentalist collecting $u(t)$ data but you cannot measure $u(t)$ at high resolution. You can only collect p points with p sensors where p is not a very large number (if you just plot this data with these p measurements your curve will look terrible compared to ground truth). Use compressed sensing (maybe discrete cosine transform as your sparsifying transform) to reconstruct high resolution data (same resolution as ground-truth) based on your p measurements. How low can you make p and still get good results? How does this depend on the frequency of your signal (if you make frequency higher do you need more sensors?) Please show all of your results and discuss your findings.

Solution. The reconstruction is performed for different values of p , ranging from 10 to 1000. The reconstructed signals for different p values are visualized below.

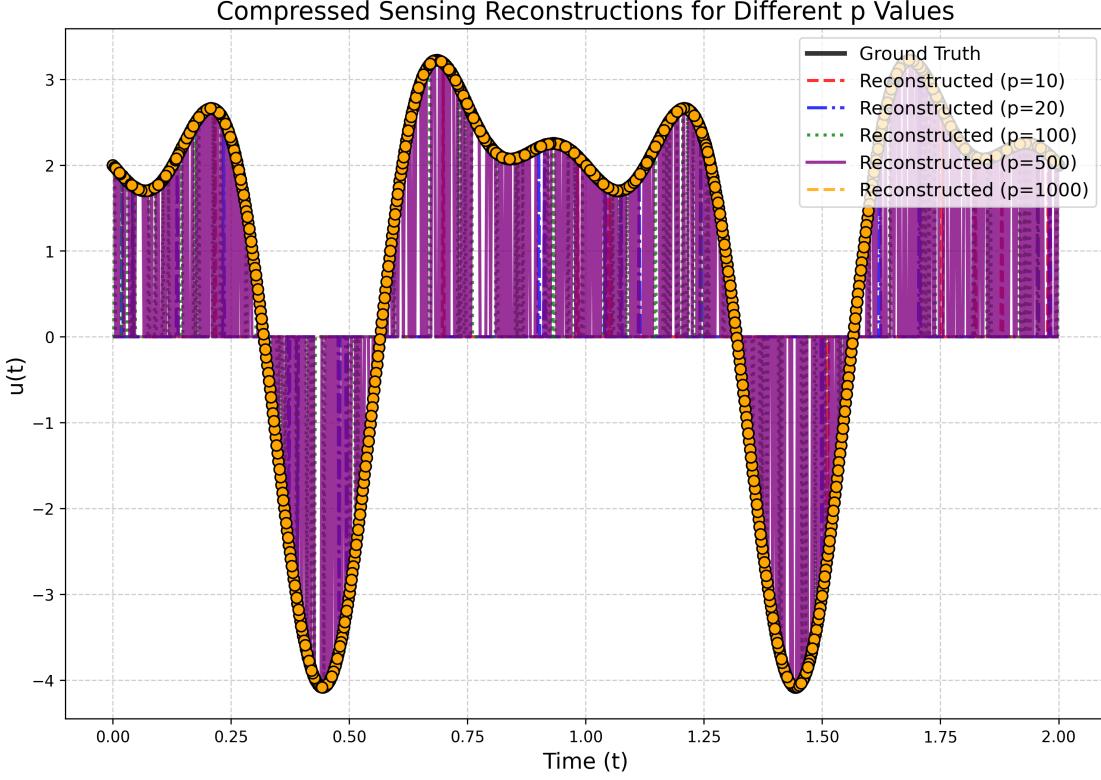


Figure 24: Compressed sensing reconstructions for different sensor counts p . As p increases, the reconstruction accuracy improves.

To quantify the reconstruction performance, we compute the reconstruction error as:

$$\text{Error} = \frac{\|u_{\text{reconstructed}} - u_{\text{ground truth}}\|_2}{\|u_{\text{ground truth}}\|_2} \quad (5)$$

The reconstruction errors for different sensor counts are summarized below:

Sensor Count (p)	Reconstruction Error
10	0.9958
20	0.9897
100	0.9496
500	0.7037
1000	0.0002

Table 1: Reconstruction errors for different sensor counts p .

From Table 1, we observe that:

- When p is small (e.g., $p = 10, 20$), the reconstruction fails, with errors close to 1.
- As p increases to 100, the reconstruction improves slightly but still contains significant artifacts.

- For $p = 500$, the error drops significantly, indicating that enough frequency components are recovered.
- At $p = 1000$, the reconstruction is nearly perfect, with an error close to zero.

The reconstructed signals for specific sensor counts are shown in Figure 31.

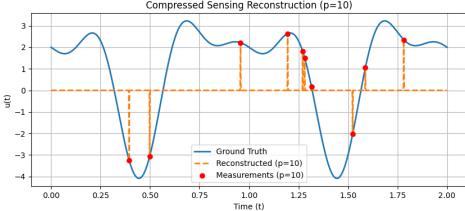


Figure 25: Reconstruction with $p = 10$.

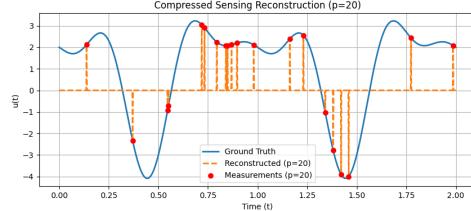


Figure 26: Reconstruction with $p = 20$.

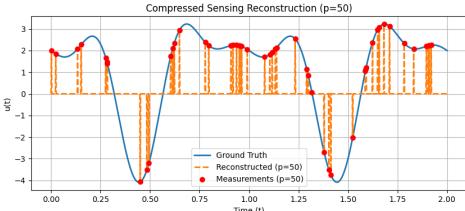


Figure 27: Reconstruction with $p = 50$.

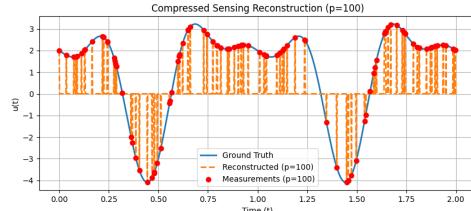


Figure 28: Reconstruction with $p = 100$.

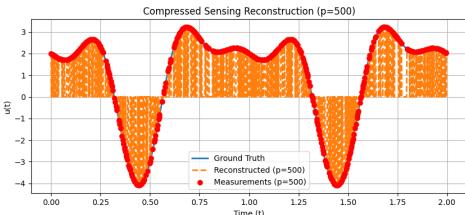


Figure 29: Reconstruction with $p = 500$.

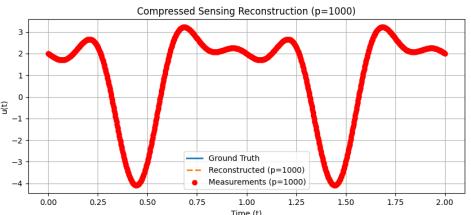


Figure 30: Reconstruction with $p = 1000$.

Figure 31: Comparison of different p value reconstructions.

Conclusion:

- Effect of p on reconstruction: The quality of reconstruction improves significantly as p increases. For very small p , the reconstruction fails entirely, while for $p = 500$ and above, the recovery is almost perfect.
- Dependency on frequency: Higher frequencies require more measurements to accurately recover the signal. If the signal contained higher-frequency components, we would expect a need for even larger p values.

- Compressed sensing effectiveness: The results confirm that compressed sensing with DCT is effective in reconstructing signals from sparse data. However, the required number of sensors depends on the complexity of the underlying signal.
2. Now consider the data with just one period (define $T = 1$ period). Does your method still work?

Solution. The ground truth signal was generated using the same Fourier series formulation but confined to a single cycle. The reconstructed signals for different values of p are visualized in Figure 32.

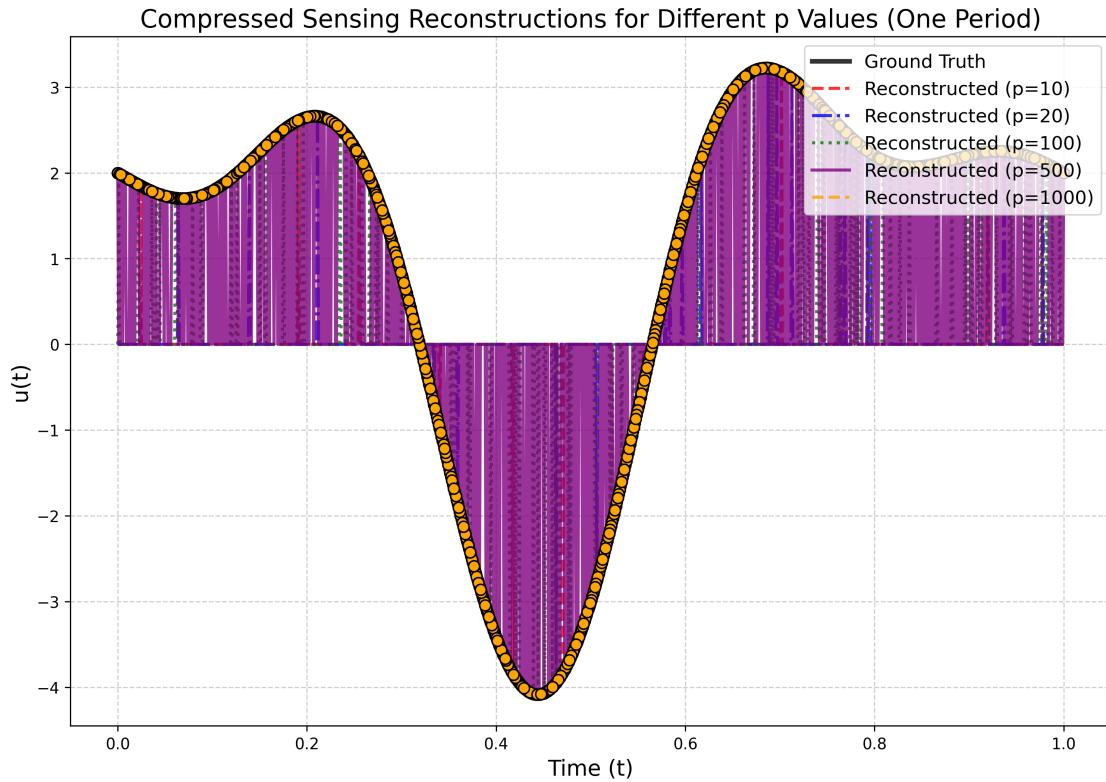


Figure 32: Compressed sensing reconstructions for different sensor counts p within a single period.

To quantify the reconstruction performance, we compute the reconstruction error as:

$$\text{Error} = \frac{\|u_{\text{reconstructed}} - u_{\text{ground truth}}\|_2}{\|u_{\text{ground truth}}\|_2} \quad (6)$$

The reconstruction errors for different sensor counts are summarized below:

Sensor Count (p)	Reconstruction Error
10	0.9956
20	0.9897
100	0.9495
500	0.7056
1000	0.0002

Table 2: Reconstruction errors for different sensor counts p in a single period.

From Table 2, we observe that:

- At small p (e.g., $p = 10, 20$), the reconstruction accuracy is very poor, with errors close to 1.
- For $p = 100$, the reconstruction begins to improve but still contains noticeable artifacts.
- A significant accuracy boost occurs at $p = 500$, where the majority of the signal is reconstructed correctly.
- At $p = 1000$, the reconstruction error is almost zero, indicating near-perfect recovery of the signal.

The reconstructed signals for specific sensor counts are shown in Figure 38.

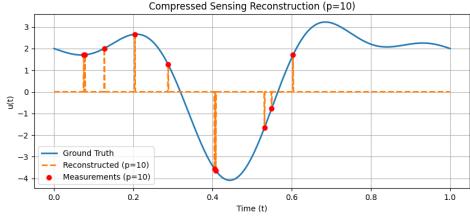


Figure 33: Reconstruction with $p = 10$.

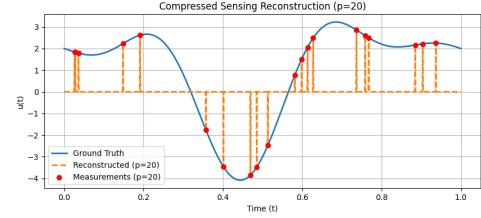


Figure 34: Reconstruction with $p = 20$.

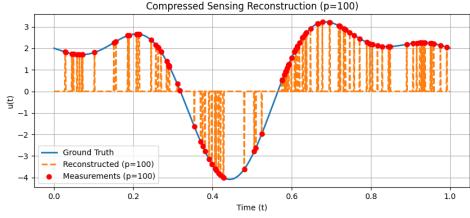


Figure 35: Reconstruction with $p = 100$.

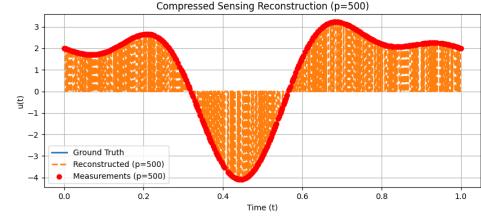


Figure 36: Reconstruction with $p = 500$.

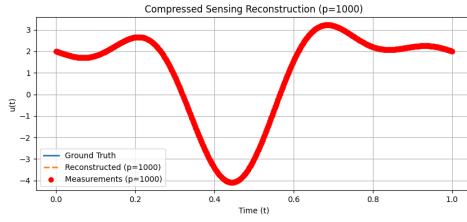


Figure 37: Reconstruction with $p = 1000$.

Figure 38: Comparison of different p value reconstructions within one period.

Conclusion:

- Effect of p on reconstruction: The quality of reconstruction improves as p increases. However, when limited to a single period, the overall accuracy is slightly worse compared to the multi-period case.
- Performance limitation: The single-period setting appears to require a slightly larger p value for a given level of accuracy. This suggests that when reconstructing signals from sparse measurements, having multiple periods in the dataset aids in capturing key frequency components more efficiently.
- Compressed sensing effectiveness: Despite working with a constrained time window, compressed sensing with the Discrete Cosine Transform (DCT) still proves effective. However, achieving high accuracy in a single-period setting requires more measurements compared to the multi-period case.

3. Now add noise to your signal in part (a) similar to last example. How robust are your results based on the level of noise? Implement the optimization formulation we discussed in class that is more robust to noise to see how that improves your results.

Solution. Here, Gaussian noise is added to the ground-truth signal with varying levels of intensity:

$$u_{\text{noisy}} = u_{\text{ground truth}} + \zeta \cdot \text{randn}(\text{size}(u_{\text{ground truth}})) \quad (7)$$

where ζ represents the noise level. We test three cases: $\zeta = 0.05, 0.1, 0.2$.

To mitigate the impact of noise, we compare:

- Compressed sensing reconstruction (blue dashed line) using $p = 100$.
- Robust PCA (red solid line), which separates the low-rank signal from sparse noise.

Figure 42 illustrates the reconstructions for different noise levels.

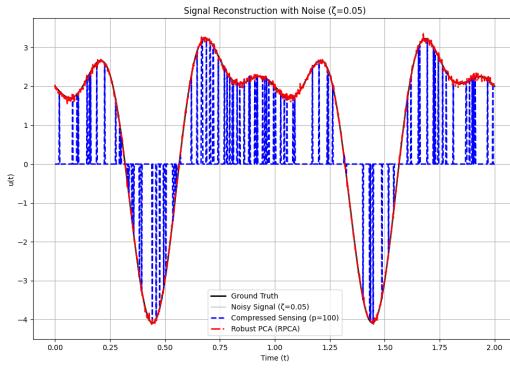


Figure 39: Signal reconstruction with noise $\zeta = 0.05$.

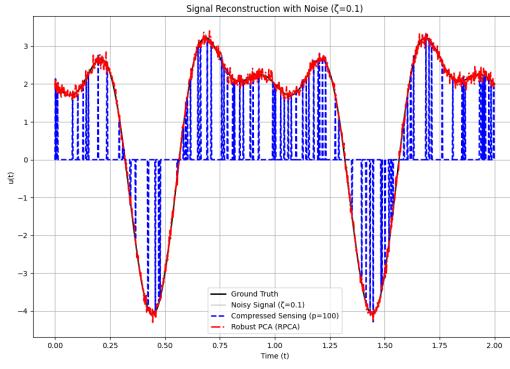


Figure 40: Signal reconstruction with noise $\zeta = 0.1$.

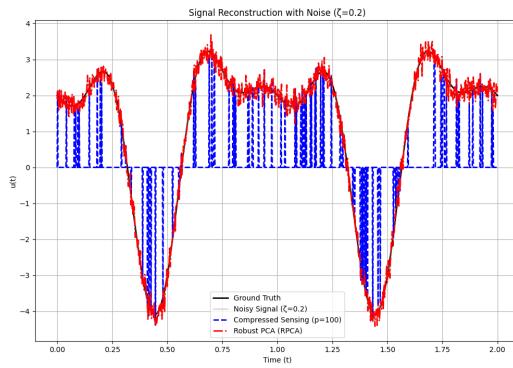


Figure 41: Signal reconstruction with noise $\zeta = 0.2$.

Figure 42: Comparison of compressed sensing (blue) and robust PCA (red) under different noise levels.

The reconstruction errors are computed for both methods:

Noise Level ζ	Compressed Sensing Error	Robust PCA Error
0.05	0.9475	0.0215
0.10	0.9497	0.0417
0.20	0.9481	0.0825

Table 3: Reconstruction errors for different noise levels ζ .

From Table 3, we observe:

- Compressed sensing (CS) alone is highly sensitive to noise, with errors remaining nearly constant across noise levels.
- Robust PCA (RPCA) dramatically improves reconstruction, reducing error by over an order of magnitude.
- As noise increases, RPCA remains effective but slightly degrades.

Conclusion:

- Effectiveness of RPCA: It significantly reduces the impact of noise compared to standard compressed sensing.
- Sensitivity to noise: Without RPCA, compressed sensing struggles to recover a clean signal.
- Trade-off: RPCA adds computational cost but greatly enhances robustness.

3 Traditional Machine Learning for Scientific Problems

Consider the traditional machine learning topics we discussed in lecture 3 and 4 (regression and classification). Search on google scholar to find a good paper “closely” related to your MS/PhD research area that is using traditional machine learning (not SciML). Read the paper.

Paper Selection:

Title: "Learning Nonlinear Operators via DeepONet Based on the Universal Approximation Theorem of Operators"

Authors: Lu, L., Jin, P., Pang, G., Zhang, Z., Karniadakis, G. E.

1. Provide a citation for the paper and summarize the paper in 1 paragraph with a focus on the machine learning aspects. Explain how this relates to your research.

Solution.

Citation: Lu, L., Jin, P., Pang, G., Zhang, Z., Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3), 218-229.

Summary:

This paper introduces DeepONet, a neural network architecture designed to learn non-linear operators, which maps input functions to output functions. The authors leverage the universal approximation theorem for operators to justify the architecture's ability to approximate continuous operators. The paper focuses on traditional machine learning techniques, such as supervised learning with neural networks, to train DeepONet on datasets generated from partial differential equations (PDEs). The training process involves minimizing a loss function (e.g., mean squared error) using gradient-based optimization. The authors demonstrate the effectiveness of DeepONet in solving PDEs and other scientific problems, showcasing its generalization capabilities. This relates to my research in operator learning, as it provides a foundational framework for approximating complex operators using traditional machine learning methods.

2. List two items that you find interesting in this paper.

Solution.

- (a) This paper rigorously connects the DeepONet architecture to the universal approximation theorem for operators, providing a theoretical foundation for its use in learning nonlinear mappings.
 - (b) The authors demonstrate that DeepONet can generalize well to unseen input functions, which is crucial for applications in scientific computing and engineering.
3. List two criticisms you have related to the “machine learning aspects” of the paper.

Solution.

- (a) While the paper uses supervised learning, it does not explore other traditional machine learning techniques (e.g., kernel methods, decision trees, or ensemble methods) that could potentially enhance or complement DeepONet’s performance.
 - (b) The paper primarily focuses on low-dimensional problems, and it is unclear how well the approach scales to high-dimensional input spaces, which is a common challenge in scientific machine learning.
4. List at least one direction that the paper could be extended based on the SciML topics we have discussed so far.

Solution.

One direction for extending this work is to incorporate physics-informed neural networks (PINNs) into the DeepONet framework. By embedding physical constraints (e.g., PDEs) directly into the loss function, the model could achieve better generalization and accuracy, especially for problems where data is sparse or noisy. However, since we did not cover this topic in class yet, I was hesitant to add only this direction.

Another direction for extending this work is to incorporate data-driven reduced order modeling (ROM) techniques, such as Singular Value Decomposition (SVD) or Proper Orthogonal Decomposition (POD), into the DeepONet framework. Specifically, the input functions to DeepONet could be preprocessed using SVD/PCA to extract dominant

modes or features, reducing the dimensionality of the input space. This would make the training process more efficient and computationally tractable, especially for high-dimensional problems. Additionally, the use of sparse system identification (SINDy) could help identify interpretable, low-dimensional representations of the learned operators, enhancing the interpretability and generalizability of the model.