Improving Koopman-Based Sequential Multifactor Disentanglement with Single Static Mode and Latent Space Refinement

Amos Haviv Hason

Supervised by **Dr. Omri Azencot**

Thesis Dissertation for the Degree of **Master of Science**
at the Department of **Computer Science**

Ben-Gurion University of the Negev

December 2024

Introduction

Representation Learning

- Focuses on learning useful features directly from raw data.
- Aims to enhance downstream tasks like classification or prediction.
- Creates a compact, meaningful latent space representation.

Sequential Disentanglement

- Disentangled representation:
  - Features map to distinct, interpretable factors (e.g., shape, color, dynamics).
  - Static and dynamic factorization.
  - Aids generalization in machine learning tasks.
- Challenges:
  - Limited labeled data.
  - Need for unsupervised solutions.
- Structured Koopman Disentanglement (SKD) model:
  - Autoencoder architecture.
  - Uses Koopman theory and dynamic mode decomposition (DMD) in bottleneck.

Our Contributions

- **Identified and addressed issues in SKD implementation.**
- **Introduced Single Static Mode Structured Koopman Disentanglement (SSM-SKD) model.**
- Proposed a greedy latent space exploration algorithm.
- **Evaluated SSM-SKD on four datasets and compared to SKD.**
- Suggested a new standard for comprehensive environment reporting to improve reproducibility.

Background

Koopman Theory and DMD

- Koopman theory:
  - Provides a linear representation of nonlinear dynamical systems through the Koopman operator.
  - Operates in an infinite-dimensional space of observables, mapping system measurements forward in time.
  - Does not linearize the system but transforms its dynamics into a linear framework for analysis.
  - Focuses on the spectral properties of the operator, including eigendecomposition, to analyze long-term behavior.

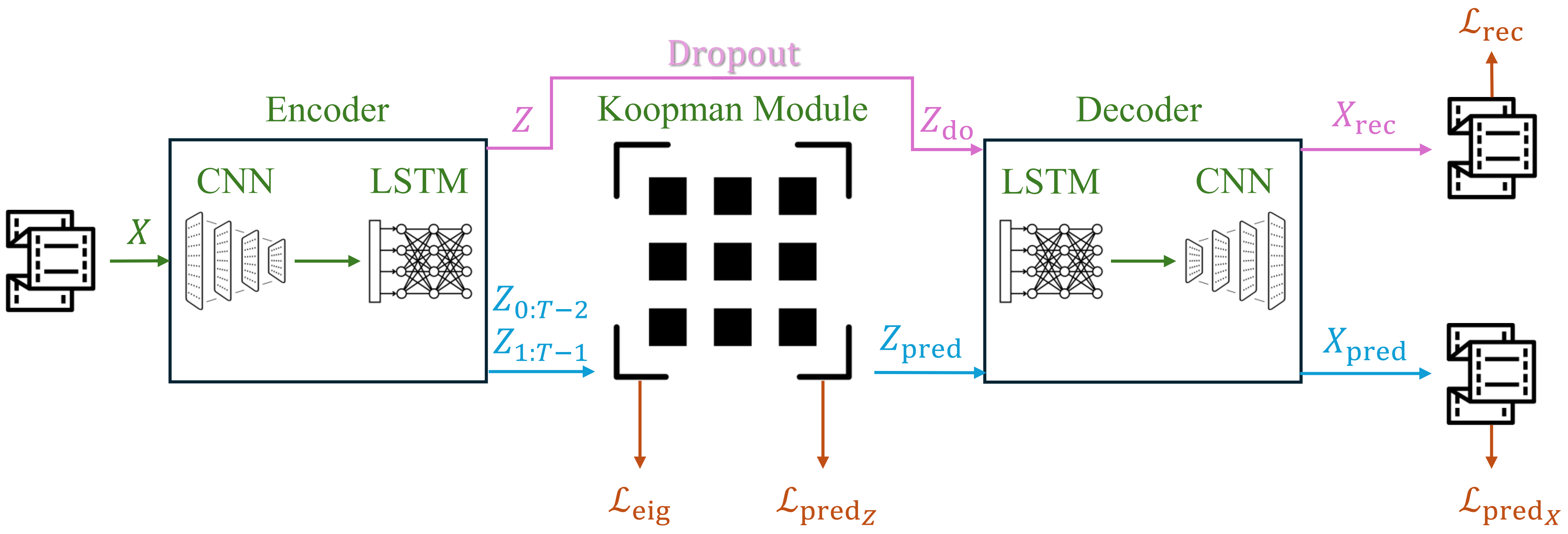
Koopman Theory and DMD

- DMD:
  - A numerical algorithm that approximates the Koopman operator from data.
  - Analyzes snapshots of a system over time to compute a finite-dimensional representation.
  - Identifies dominant dynamic modes and their eigenvalues (frequencies, growth/decay rates).
  - Initially developed for fluid mechanics, it is widely used for high-dimensional, time-dependent data.

Koopman Theory and DMD

- Relationship:
  - Koopman theory is a theoretical framework for understanding nonlinear dynamics via linear operators.
  - DMD is a practical application that approximates the Koopman operator from finite, observable data.
- Applications:
  - Fluid mechanics, video processing, time-series analysis, and system identification.
  - Enable higher-level insights on the behavior of dynamical systems.

SKD



* Icons created by iconsmind.com, Oleksandr Panasovskyi, Lucas Rathgeb, and ahmadwil from Noun Project

SKD

$$\mathcal{L}_{\text{pred}_Z} = \text{MSE}(Z_{\text{pred}}, Z)$$

$$\mathcal{L}_{\text{pred}_X} = \text{MSE}(X_{\text{pred}}, X)$$

$$\mathcal{L}_{\text{rec}} = \text{MSE}(X_{\text{rec}}, X)$$

$$\mathcal{L}_{\text{eig}} = \frac{1}{|S|}\sum_{\lambda \in S} |\lambda - 1|^2 + \frac{1}{|D|}\sum_{\lambda \in D} \begin{cases} \text{Re}(\lambda), & \text{if } \text{Re}(\lambda) > \alpha \\ 0, & \text{otherwise} \end{cases}$$

$$\mathcal{L} = w_{\text{pred}_Z}\mathcal{L}_{\text{pred}_Z} + w_{\text{pred}_X}\mathcal{L}_{\text{pred}_X} + w_{\text{rec}}\mathcal{L}_{\text{rec}} + w_{\text{eig}}\mathcal{L}_{\text{eig}}$$

Reimplementing SKD

- **Motivation:**
  - **Reproducing SKD results was hindered by multiple issues in the original implementation.**
- Dimension mismatch in architecture:
  - Original implementation misused two dimension hyperparameters.
  - Fixed by aligning code with Table 5 of the SKD paper.
- Inconsistent size of subset *S* across batches:
  - Usage of function get_unique_num() for delimiting static and dynamic eigenvalues caused spectral loss instability.
  - Resolved by consistently considering *s* eigenvalues closest to 1 as static-related, regardless of conjugate pairing.
- NaN gradient issues:
  - Training often failed (>= 40% of runs) due to NaN values in gradients.
  - Addressed by applying gradient clipping for numerical stability.

Reimplementing SKD

- Precision in eigenvalue computations:
  - Original implementation used lower precision operations.
  - Upgraded to float64 for Koopman module and spectral loss calculation to improve numerical stability.
- Learning rate scheduling:
  - Original lacked a learning rate scheduler, leading to suboptimal convergence.
  - Added a scheduler to decay learning rate on plateau.
- Hyperparameter discrepancies (Sprites dataset):
  - Reported hyperparameters did not reproduce results.
  - Adjusted.
- **Impact:**
  - **These corrections enhance SKD's reproducibility, stability, and convergence, laying a robust foundation for further experimentation.**

Single Static Mode Structured Koopman Disentanglement (SSM-SKD)

Motivation

- In SKD, static modes are constrained to have eigenvalues ~1.
- SSM-SKD reduces all static modes to a single static mode with an eigenvalue ~1.
- Potential benefits:
  - Allows tighter constraints on static modes.
    - Current deep learning software platforms do not support backpropagation through eigenvectors due to numerical difficulties.
    - It is possible to approximate an eigenvector related to a real eigenvalue using a backpropagation-friendly algorithm.
  - Simplifies static mode representation and static disentanglement.
    - Orthogonality in coordinates.

Architecture

- With $s = 1$ (one static eigenvector), SKD faces the **shortcut problem** (disentanglement-reconstruction tradeoff):
  - Low K (Koopman operator size) values ($K <= 8$): Poor reconstruction performance.
  - High K values: Poor static-dynamic disentanglement as the model encodes static information in other modes (they have more capacity).
- Instance-wise Koopman operator approximation:
  - Replace batch-level Koopman operator approximation with instance-level approximation.
  - Solve least squares problem for each instance.

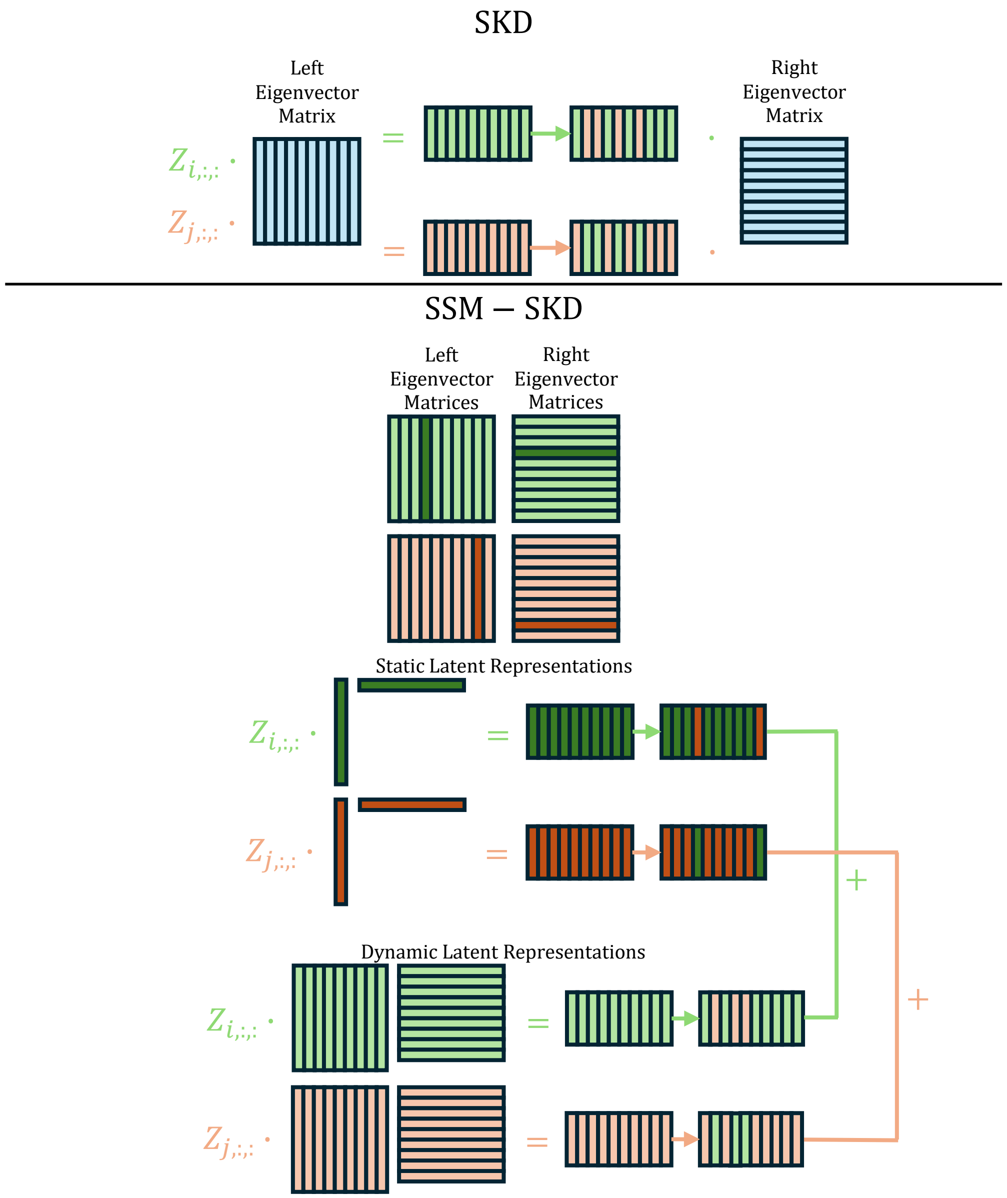
Attribute Swapping in SKD

- Latent space extraction:
  - SKD computes a latent space per batch.
  - Koopman latent representation for instance:
    - Z_i × (desired eigenvector submatrix of the Koopman operator)
- Attribute swapping process:
  - Multiply latent matrices by eigenvector matrix.
  - Swap desired modes between instances in the resulting matrices.
  - Multiply swapped matrices by the inverse of the eigenvector matrix to obtain new latent matrices.
  - Decode the modified latent matrices for swapped outputs.

Attribute Swapping in SSM-SKD

- Instance-wise Koopman operator approximation:
  - Each instance has its own Koopman operator.
  - SKD’s batch-based method is incompatible with SSM-SKD.
- New method:
  - Compute static latent representation:
    - Z_i × (static mode submatrix of eigenvector matrix) × (submatrix of eigenvector inverse)
  - Compute dynamic latent representation similarly with dynamic modes.
  - Treat coordinates of static and dynamic representations as channels.
  - Swap desired channels between instances.
  - Sum static and dynamic representations, then decode.
- Lacks theoretical guarantees or justification.
  - Contrary to SKD’s approach which is grounded on DMD.

Comparison



SKD
Left Eigenvector Matrix
Right Eigenvector Matrix
$Z_{i,:,:}$ ·
$Z_{j,:,:}$ ·
SSM – SKD
Left Eigenvector Matrices
Right Eigenvector Matrices
Static Latent Representations
Dynamic Latent Representations

Latent Space Exploration

- Which static modes relate to each factor?
  - Partition of channels to factor sets.
- Train classifiers for static factors of the desired dataset.
- Sample instances from dataset and swap latent channels between them.
- SKD employs a brute-force search over the power set of static modes.
- We propose a greedy latent space exploration algorithm.
  - Swap all channels except a single channel.
  - Tie channel to the static factor for which accuracy is maximal.
  - We prove that for our coordinate-based approach, it yields an optimal solution regarding the **sum of factor accuracies**.
  - Does not minimize **leakage between factors**.
    - Leakage: Information about a factor being located in channels which are tied other factors.

Evaluation

Metrics

$$\mathcal{D}_{\text{sd}}(X) = 1 - \frac{1}{2(|F_{\text{s}}| + |F_{\text{d}}|)} \Bigg(\sum_{f \in F_{\text{s}}} \left| \text{Acc}(\mathbb{C}_f(\text{StaticSampleSwap}(X)), \mathbb{C}_f(X)) - \frac{1}{|\mathcal{C}_f|} \right| + \\ \sum_{f \in F_{\text{d}}} \left| \text{Acc}(\mathbb{C}_f(\text{StaticSampleSwap}(X)), \mathbb{C}_f(X)) - 1 \right| + \\ \sum_{f \in F_{\text{s}}} \left| \text{Acc}(\mathbb{C}_f(\text{DynamicSampleSwap}(X)), \mathbb{C}_f(X)) - 1 \right| + \\ \sum_{f \in F_{\text{d}}} \left| \text{Acc}(\mathbb{C}_f(\text{DynamicSampleSwap}(X)), \mathbb{C}_f(X)) - \frac{1}{|\mathcal{C}_f|} \right| \Bigg)$$

$$\mathcal{D}_{\text{mf}}(X) = 1 - \frac{1}{|F_{\text{s}}|(|F_{\text{s}}| + |F_{\text{d}}|)} \left(\sum_{f \in F_{\text{s}}} \sum_{g \in F_{\text{s}} \cup F_{\text{d}}} \begin{cases} \left| \text{Acc}(\mathbb{C}_g(\text{FactorialSampleSwap}_f(X)), \mathbb{C}_g(X)) - 1 \right|, & \text{if } g = f \\ \left| \text{Acc}(\mathbb{C}_g(\text{FactorialSampleSwap}_f(X)), \mathbb{C}_g(X)) - \frac{1}{|\mathcal{C}_g|} \right|, & \text{otherwise} \end{cases} \right)$$

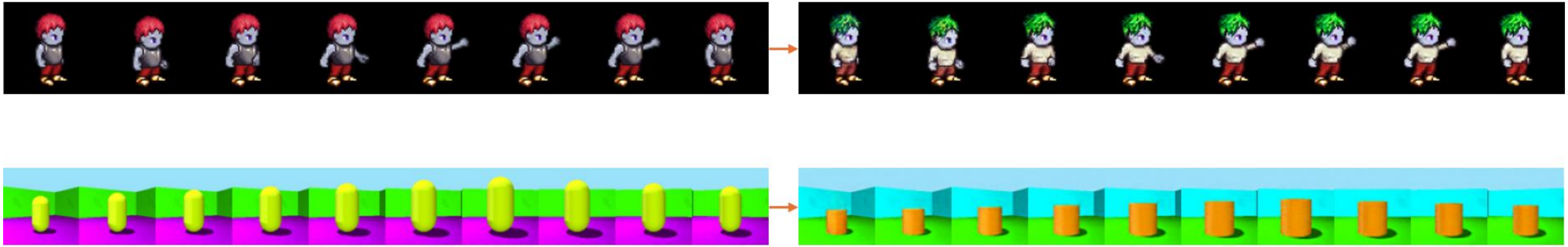
Static-Dynamic Disentanglement Results

Swap	Color	Shape	Scale	Position X	Position Y
Static	0.1902	0.8488	**0.8857**	**0.9827**	**0.9939**
Dynamic	**0.9976**	**0.4907**	0.1301	0.1721	0.1605

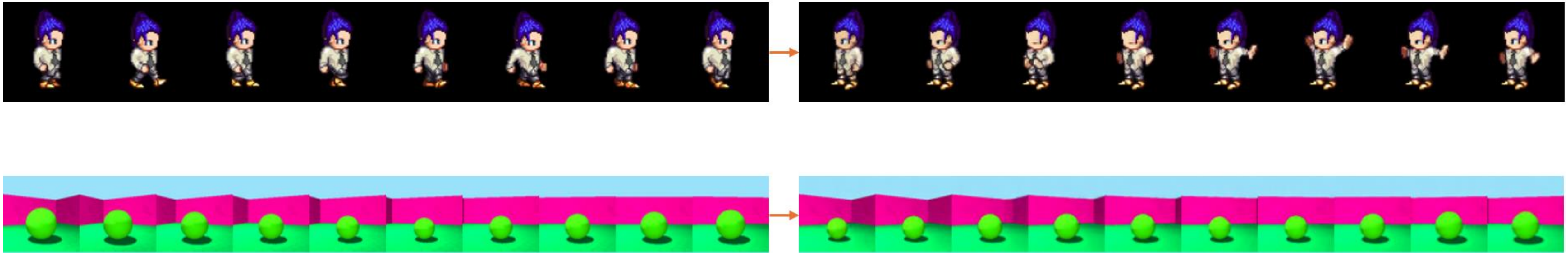
Sprites	dSprites	Moving dSprites	3D Shapes
0.9872	0.9491	0.8699	0.9492

Static-Dynamic Disentanglement Results

Static Sample Swap



Dynamic Sample Swap



Multifactor Disentanglement Results

Retain	**Color**	**Shape**	**Scale**	**Position X**	**Position Y**
Color	**0.9845**	0.3431	0.1043	0.1301	0.1315
Shape	0.18	**0.4738**	0.1268	0.1485	0.1465

Sprites	**dSprites**	**Moving dSprites**	**3D Shapes**
0.9836	0.9142	0.9348	0.971

Alternative Metrics

- Current metrics are non-sensitive to weak local performance.
  - All scores are close to 1, even in cases of weak performance.
  - The range of values between 0 and 1 is not used efficiently.
  - Uninformative.
- Measure distance between actual accuracy and target accuracy on a linear scale from 0 to 1.
- Use geometric mean instead of arithmetic mean.
- This was the original approach.
  - Replaced by current one for simplicity.

Discussion

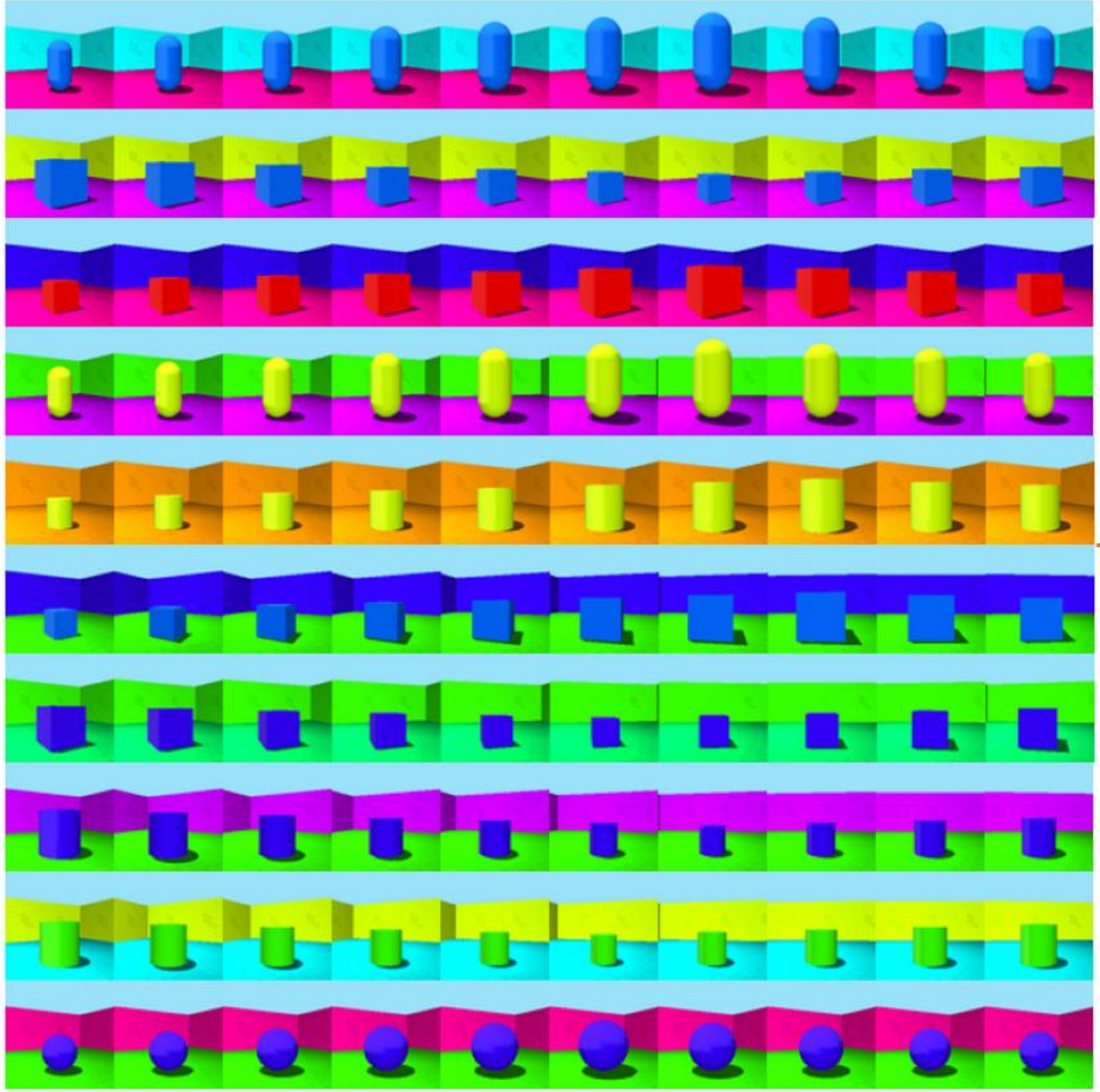
Comparison with SKD on Sprites

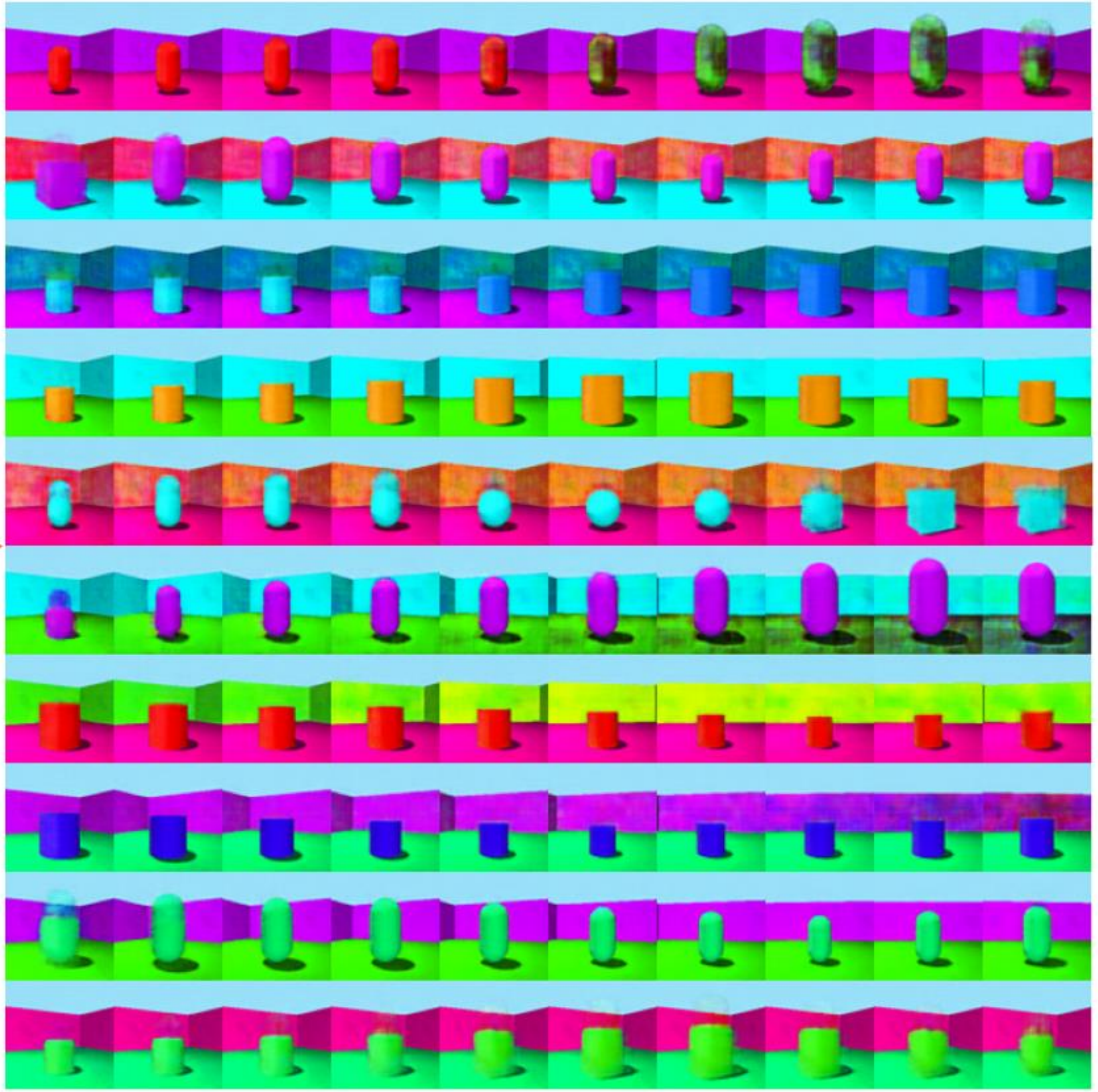
- Metrics:
  - Static-dynamic disentanglement:
    - SKD: 0.9981
    - SSM-SKD: 0.9872 (-0.0109)
  - Multifactor disentanglement:
    - SKD: 0.9276
    - SSM-SKD: **0.9836 (+0.056)**
- Both models: ~2M parameters.
- Latent space size:
  - SKD: $K = 40$
  - SSM-SKD: **K = 15 (2.667x smaller)**

Limitations

- Theoretical gaps:
  - No formal justification for latent space extraction method.
- Dataset splits:
  - Model selection uses test data, risking overfitting.
- Inconsistency between frames:
  - Static factors may vary across frames.
  - Sequence-level classifiers overcome this during evaluation.
  - Need for frame-level evaluation metrics.
- Poor performance on dSprites variants:
  - Fails to disentangle shapes from dynamics, especially on Moving dSprites.

Inconsistency Between Frames





Future Work

- Multifactor disentanglement of dynamics:
  - Preliminary results on 3D Shapes show potential.
- Ablation study:
  - Compare SSM-SKD to SKD with new coordinate-based latent space extraction method across datasets.
- Explore using the shifted inverse power method for eigenvector approximation to introduce constraints on static latent representations.
- Establish robust datasets and metrics for sequential multifactor disentanglement benchmark.
- Study failure cases on dSprites variants.
- Explore sequential multifactor disentanglement in domains other than vision.

Questions?

Thank you!