

Article

Latent Reorganization by Fixed-Point Regularization in Recurrent Frame Prediction

Arpan Ghosh , Jinho Kim , Ye-Chan An  and Tae-Yong Kuc * 

Department of Electrical and Computer Engineering, College of Information and Communication Engineering, Sungkyunkwan University, Suwon 16419, Republic of Korea; ghosharpan@skku.edu (A.G.); jinho9812@g.skku.edu (J.K.); ycan@g.skku.edu (Y.-C.A.)

* Correspondence: tykuc@skku.edu

Abstract

Regularization can improve generalization by constraining how a model uses its internal representation. In this paper, we study whether algebraic fixed-point constraints applied to the final LSTM hidden state during training can reorganize the recurrent latent space and improve held-out frame prediction. Rather than modifying the inference-time architecture, we introduce four training-time operators, GlobalHouseholder (reflection), GlobalGivens (rotation), Composition, and Lie Algebra, that bias the hidden state toward geometrically structured regions without changing the decoder pathway. Experiments across four datasets (indoor robot sequences, KITTI driving, Flying Shapes 2D, and Moving 3D Shapes) show that lightweight constraints consistently improve prediction on structured scenes, with GlobalGivens achieving up to +1.04 dB PSNR and −11.3% MAE over the unconstrained baseline on held-out Indoor sequences. The latent analysis reveals that the operators that generalize best are not those that compress the representation most aggressively but those that redistribute latent energy while preserving broad dimensional participation. Lie Algebra, despite collapsing activation variance by 81–96%, degrades under latent perturbation and does not match the lighter operators on structured datasets identifying over-constraint as a clear failure mode. These results suggest that geometric regularization of a recurrent bottleneck can act as a useful training-time prior without adding any inference overhead.

Keywords: fixed-point regularization; latent reorganization; recurrent neural networks; video prediction; geometric deep learning; latent geometry; generalization



Academic Editor: John A. Kalomiros

Received: 27 June 2026

Revised: 20 July 2026

Accepted: 21 July 2026

Published: 25 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The success of deep learning in computer vision [1–3], natural language processing [4–6], and electronics-related design tasks [7–9] is often attributed to its ability to learn richer internal representations [10,11]. This view is consistent with the manifold hypothesis, which suggests that high-dimensional observations often concentrate near lower-dimensional geometric structures [12–16]. Geometric properties such as symmetry and permutation invariance can be used as inductive biases to improve representation learning. For example, graph neural networks are represented by many equivalent node orderings, showing their permutation-invariant nature [17,18]. This motivates a broader view in which held-out prediction can improve when arbitrary coordinate dependence is reduced or reorganized within the representation [19]. For example, a permutation-invariant graph neural network improves generalization by forcing the latent code to represent structure

independently of node ordering [20]. In a set autoencoding framework, permutation-invariant encoders force the latent state to encode the content of an unordered collection rather than its input order [21]. Similarly, permutation-invariant discrete representation learning suggests that removing positional identity from latent codes can encourage more global and semantically organized representations [22]. Figure 1 illustrates a representative example of this indoor visual-sensor data-collection setting, showing the floor-map layout together with the robot platform and corresponding RGB scenes; the arrow indicates the robot's direction of travel through the corridor.

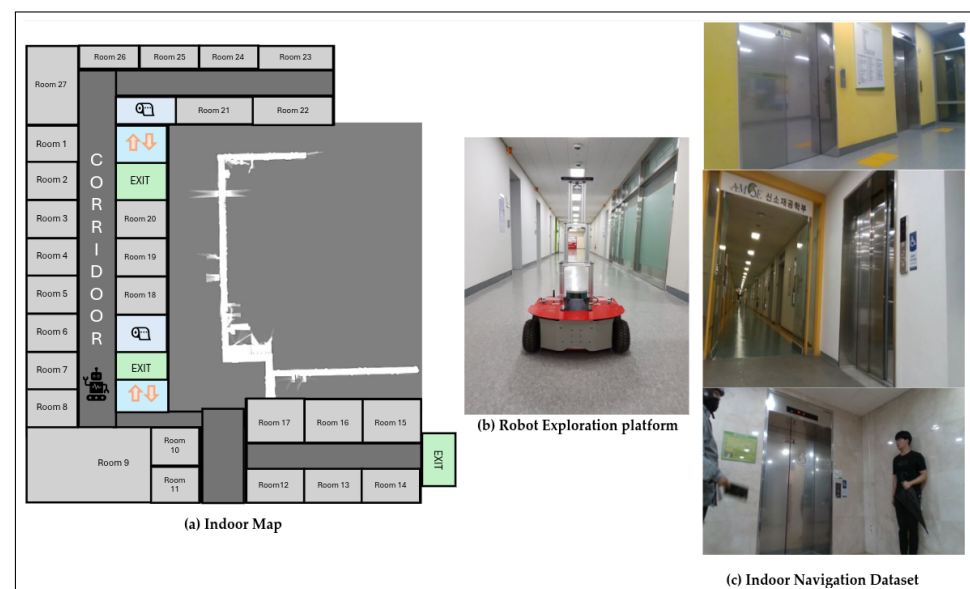


Figure 1. Representative indoor visual-sensor data-collection environment. A mobile robot platform is shown operating in a structured university corridor, together with the corresponding floor-map layout and RGB examples from corridor and elevator areas. The robot setting motivates the use of compact recurrent visual prediction, where future-frame estimation depends on temporal sensor observations collected under viewpoint changes, repeated architectural structures, and partial scene symmetry.

The indoor dataset is motivated by mobile robot perception in structured building environments. During navigation, a robot repeatedly observes corridors, elevator doors, wall boundaries, signage, and partially symmetric architectural layouts from changing viewpoints. These visual-sensor streams provide a practical setting for studying recurrent frame prediction because the model must preserve temporal information about scene layout while remaining robust to local appearance variation. In this work, the robot environment is used as a representative source of structured visual sequences; the focus is on the latent organization induced by fixed-point regularization rather than on closed-loop robot control. No claim is made about autonomous navigation performance; the robot platform is used to obtain structured visual-sensor streams for controlled latent-geometry evaluation.

Existing next-frame prediction methods typically improve performance by increasing architectural capacity: stronger recurrent units, spatial memory modules, attention mechanisms, skip connections, or transformer-based temporal modeling [23–26]. These approaches are effective, but they do not isolate the role of the final latent state itself. Unitary and orthogonal recurrent networks impose algebraic constraints on recurrent weight matrices to stabilize gradients [27,28], while group-equivariant networks encode geometric structure in spatial feature maps [29,30]. In contrast, our goal is not to build a larger predictor nor to impose equivariance on the image plane. We directly regularize the geometry of the recurrent bottleneck.

We use next-frame prediction as a controlled probe for studying latent geometry. Given a sequence of past frames, the model predicts the immediate next frame, but reconstruction sharpness is not the main object of study. Instead, the prediction task provides a supervised setting in which we test whether algebraic fixed-point regularization changes the organization and robustness of the recurrent hidden state. Rather than applying explicit permutations to a spatial latent tensor, we impose operator-based fixed-point constraints on the final LSTM hidden state $h_7 \in \mathbb{R}^k$.

In this framework, the latent object is not a spatial tensor $z \in \mathbb{R}^{C \times H \times W}$ but the LSTM bottleneck state $h_7 \in \mathbb{R}^k$. Transformations like rotation, reflection, and manifold symmetry operators act on h_7 and reorganize the latent space. The training objective biases the hidden states toward operator-compatible regions before decoding. The formulation can in principle be applied to other encoder–decoder models with an explicit latent bottleneck, but this paper evaluates it only in a controlled recurrent setting. In addition, the information pathway between the encoder and the decoder is set through the latent space. Reorganizing h_7 changes the representation available to the decoder without changing the inference-time pathway. Architectural shortcuts, such as spatial skip connections, are not used in this paper to create additional information pathways. The aim is not to improve visual sharpness when predicting the next video frame but to examine the sensitivity of latent-space reorganization to the model’s generalizability. Rather than targeting reconstruction quality alone, our work focuses on empirical evidence for geometric latent-space reorganization in a controlled recurrent prediction setting. The indoor sequences considered in this study originate from a mobile robot observing structured corridors, elevator areas, doorway boundaries, and repeated architectural elements over time. This setting provides a practical source of recurrent visual-sensor streams, but the main purpose of the prediction task is not to produce visually sharp future frames. Instead, the task is used as a controlled probe for studying how temporal scene information is organized inside the final recurrent bottleneck state. Because the decoder receives only the compact hidden state h_7 , changes in held-out behavior can be examined together with changes in latent energy distribution, effective rank, active dimensionality, and noise response. In this context, the fixed-point operators act as training-time geometric biases on the latent state, allowing us to test whether operator-compatible organization emerges without changing the inference-time decoding path.

The experimental model consists of a 3D convolutional encoder, a single-layer LSTM-Cell, and a no-skip decoder. Given $T = 18$ input frames, the encoder and recurrent module produce the final hidden state $h_7 \in \mathbb{R}^k$, from which the decoder predicts the next frame x_{T+1} . Because no skip connections are used, all information passed from the input sequence to the decoder must flow through h_7 .

The model’s performance is evaluated using four latent diagnostic metrics: Shannon entropy, Dims@90, effective rank, and activation variance. These metrics serve as proxies for the degree of coherent geometric structure induced by each operator, without requiring direct estimation of mutual information. For the two real-world datasets, generalization is evaluated under held-out environment protocols: (i) For the Indoor dataset, training is performed using sequences from five training environments, while the held-out test set is collected from entirely separate buildings and environments. No frames from the held-out test buildings are used during training, validation, or hyperparameter selection. (ii) For KITTI, a drive-disjoint protocol holds out five drives under the same conditions. Both constitute zero-shot cross-environment evaluation settings in which the model has no access to any appearance statistics of the test environment during training.

The machine learning model is trained to predict the next frame. First, from an input sequence $x_{1:T}$, the encoder and the recurrent module define a mapping

$$h_7 = F_\theta(x_{1:T}) \in \mathbb{R}^k,$$

where h_7 is the final LSTM hidden state. The encoder E_θ maps an input image sequence $x_{1:T}$ to a latent representation h_7 , and the decoder predicts the next frame as:

$$\hat{x}_{T+1} = D_\phi(h_7).$$

Here, θ and ϕ are the learnable parameters of the encoder and decoder, respectively. We have used the following latent transformation operations: $G = \{U_1, U_2, U_3, U_4\}$, where U_1 = rotation (GlobalGivens), U_2 = reflection (GlobalHouseholder), U_3 = Composition, U_4 = Lie Algebra. The algebraic operator is applied only to h_7 during training. For an operator U , the fixed point is

$$\mathcal{L}_{\text{fp}} = \|U(h_7) - h_7\|_1.$$

Here, GlobalHouseholder imposes a reflection-like fixed-point structure, GlobalGivens imposes a rotation-like constraint, Composition combines reflection and rotation, and Lie Algebra provides a more expressive continuous transformation family. Our model is then trained to minimize this fixed-point loss. For operators defined through algebraic closure, the corresponding closure loss is used, for example

$$\mathcal{L}_{\text{rot}} = \|\mathcal{R}^m(h_7) - h_7\|_1, \quad \mathcal{L}_{\text{ref}} = \|\mathcal{M}(h_7) - h_7\|_1.$$

The reconstruction objective is

$$\mathcal{L}_{\text{rec}} = \|\hat{x}_{T+1} - x_{T+1}\|_1,$$

and the total objective is

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{fp}} \mathcal{L}_{\text{fp}}.$$

The algebraic operator is used only as a training-time regularizer. At inference time, the decoder receives the untransformed hidden state h_7 . The main experiments use $\lambda_{\text{fp}} = 1.0$ for all operator variants to keep the comparison controlled. A supplementary sensitivity analysis over $\lambda_{\text{fp}} \in \{0.5, 1.0, 2.0\}$ is reported in Appendix A.5.

The fixed-point loss is used as a training-time geometric bias on the final recurrent hidden state, not as a proof of decoder invariance. For each sequence, the encoder and recurrent module produce $h_7 = F_\theta(x_{1:T}) \in \mathbb{R}^k$. The auxiliary loss penalizes the distance between h_7 and its transformed or algebraically closed counterpart, depending on the operator. This biases the learned hidden states toward operator-compatible regions of $\mathbb{R}^k$. The constraint can reduce effective representational freedom, but it does not necessarily reduce the number of active dimensions. This distinction is tested empirically in Section 4.

The main contribution of this paper is to connect symmetry-inspired representation learning with recurrent temporal prediction. Previous work has shown that permutation invariance, equivariance, and invariant latent projections can improve representation quality by removing arbitrary coordinate dependence. We extend this idea to recurrent video prediction by showing that algebraic fixed-point constraints can reorganize the LSTM's final hidden state and improve generalization under a controlled bottleneck. The results suggest that held-out prediction can improve when the recurrent state is biased toward operator-compatible regions rather than left unconstrained.

The contributions of this paper are summarized:

- We propose our fixed-point regularization of recurrent hidden states as a training-time geometric prior for temporal frame prediction.
- We show how algebraic operators (GlobalHouseholder, GlobalGivens, Composition, and Lie Algebra) reorganize the final LSTM hidden state around invariant sets without changing the inference-time forward pass.
- We show that improved held-out behavior is associated with broader, operator-compatible latent organization rather than aggressive latent concentration and identify over-constraint as a failure mode.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the controlled architecture, datasets, training protocol, and evaluation metrics. Section 4 reports and discusses the latent-geometry, robustness, operator-specific behavior, and limitations. Section 5 concludes this paper.

2. Related Work

Recently, there has been a significant surge in research on geometric deep learning, especially on the role of permutation invariance in generalization capacity. Researchers are showing that generalization arises not only from stronger architectures but also from order-dependent variation that is reorganized within the representation. Winter et al. introduce a novel framework of a permutation-invariant encoder and a differentiable sorting operator. This generates robust, lower-dimensional graph representations for improved downstream task performance [17]. Duan and Lee formulate graph generation as a permutation-invariant latent-variable problem, avoiding the need to impose a fixed sequential node order [18]. Kortvelesy et al. [21] further extend this line of work to set autoencoding. Here, an encoder summarizes an unordered collection without treating the input order as meaningful. Thiede et al. [20] develop a theoretical framework of permutation-equivariant neural networks and graph variational encoders. Hansen et al. [19] show that equivariant models still require direct analysis of the learned representations. This is because architectural equivariance alone does not fully account for how information is organized in the latent space. Cohen et al. first introduced group-equivariant convolutional networks that build transformation structure into spatial feature maps. These learned feature maps respond predictably to group actions such as rotations and translations [29]. An advanced class of equivariant architectures is used for broader transformation groups [30].

Another relevant line of work investigates invariance and consistency as training objectives. Simard et al. [31] proposed a tangent propagation method that regularizes a network by penalizing the sensitivity to known transformation directions. This encourages local invariance around each training example. On the other hand, contractive autoencoders study explicit invariance by penalizing the Jacobian encoder sensitivity [32]. Consistency regularization methods extend the same idea by requiring predictions to remain compatible under stochastic perturbations, adversarial perturbations, or data augmentations [33–36]. Algebraic operators, for example, Householder, have been used to construct efficient orthogonal recurrent transformations [28]. The algebraic structure is used within the recurrent transition itself, primarily to improve optimization stability. Our use of algebraic operators differs from this line of work. Instead, reflection, rotation, composition, and Lie-type operators are introduced as auxiliary constraints on the final hidden state during training. The inference-time model remains unchanged, so the role of the algebraic operator is to reorganize the recurrent bottleneck rather than to define the recurrent dynamics [27].

Recent research on fixed-point iterations in deep networks studies how hidden states can stabilize after looped transformations and provides theoretical conditions under which robust fixed points can arise in high-dimensional neural systems [37]. Fixed-point reasoning models use convergence toward a fixed point as a mechanism for adaptive computation or

halting in looped architectures [38]. These works are relevant to our paper in that they treat fixed points as a property of an iterative neural process. The present work also differs in that the fixed-point structure is imposed. We do not use fixed-point convergence to halt computation. Also, we do not analyze the repeated application of network layers. Instead, we impose algebraic fixed-point conditions on the final recurrent hidden state h_7 during training. In the practical setting, fixed-point quantization of convolutional networks studies how weights and activations can be represented with optimized fixed-point bit widths to reduce storage and computation [39].

Our model architecture is related to long short-term memory networks [40] and algebraically constrained recurrent models. Orthogonal and unitary recurrent networks use norm-preserving transformations to stabilize hidden dynamics and reduce gradient instability through long temporal horizons [27,28]. Householder reflections have also been used as efficient parameterizations of orthogonal transformations in recurrent networks [28]. These methods use algebraic structure inside the recurrent transition itself, mainly for optimization and stability. Our use of algebra is different. The LSTM transition is not replaced by a unitary, orthogonal, or equivariant recurrent update. Instead, reflection, rotation, composition, and Lie-type operators are introduced as auxiliary training-time constraints on the final recurrent state. The inference-time forward pass remains unchanged, so the role of the algebraic operator is to reorganize the latent bottleneck rather than to define the recurrent dynamics.

3. Materials and Methods

3.1. Controlled Model Architecture

We use a controlled encoder–recurrent–decoder architecture composed of a 3D CNN encoder, a single-layer LSTMCell, a training-time operator block, a no-skip decoder, and a prediction head, as shown in Figure 2. The architecture is intentionally not designed as a high-capacity video-prediction baseline. Instead, it is used as an identification setting in which all information passed to the decoder must strictly flow through a single flat recurrent bottleneck, $h_7 \in \mathbb{R}^k$. This makes it possible to test whether different algebraic fixed-point constraints reorganize the same latent state under matched encoder, decoder, and training conditions. The model is trained to predict the next frame using an L1 reconstruction loss, with an additional training-time auxiliary loss applied to h_7 for the regularized variants. Next-frame video prediction provides an experimental setting for testing generalizability with latent reorganization.

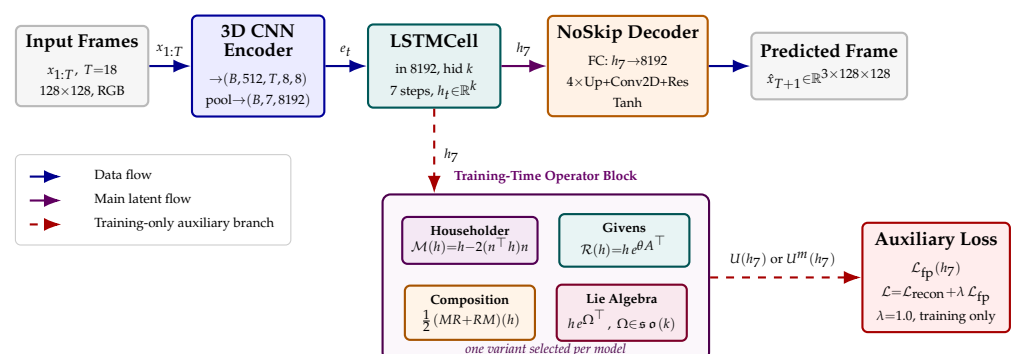


Figure 2. Overview of the controlled recurrent prediction architecture. The decoder receives the untransformed final LSTM hidden state h_7 . The algebraic operator block is used only as a training-time auxiliary branch to compute the fixed-point or closure loss on h_7 ; it is not part of the inference-time decoding path.

Table 1 summarizes the five variants of the model.

The operator parameter counts are tiny relative to the full encoder–decoder parameter count; the largest operator, Lie Algebra, adds only 2016 parameters at $k = 64$. This difference in parameterization also changes the strength of the constraint. GlobalGivens constrains h_7 through a single learnable rotation parameter, leaving most latent directions comparatively free. Lie Algebra instead applies a full $SO(k)$ rotation parameterized by $k(k-1)/2$ learnable degrees of freedom, equal to 2016 at $k = 64$. This stronger parameterization can restrict many latent directions simultaneously, which is consistent with the substantially lower activation variance observed for Lie Algebra, 81–96% across datasets, whereas Givens broadens Dims@90 on all four datasets.

Table 1. Model variants at $k = 64$. All variants share the same encoder, decoder, recurrent bottleneck size, and training protocol. The regularized models differ only in the training-time algebraic operator and its corresponding auxiliary loss.

Variant	Operator $\mathcal{S}$	Learnable Params	Auxiliary Loss
Baseline	I	0	None
Householder	$\mathcal{M}$	k	$\mathcal{L}_{\mathcal{M}}$
Givens	$\mathcal{R}$	1	$\mathcal{L}_{\mathcal{R}}$
Composition	$\mathcal{C}$	$k+1$	$\mathcal{L}_{\mathcal{C}}$
Lie Algebra	$\mathcal{R}_{\Omega}$	$k(k-1)/2$	$\mathcal{L}_{\Omega}$

We included a ConvLSTM-cell control as a lightweight surrogate reference for spatiotemporal recurrent modeling. It preserves the same encoder, no-skip decoder, $k = 64$ bottleneck interface, optimizer, and evaluation protocol while replacing only the flat LSTM-Cell with a convolutional recurrent cell. Thus, the comparison serves as a recurrent-cell ablation under the same bottleneck-identification setting, rather than as a full architectural benchmark.

To verify that these operators do not introduce inference-time computational overhead, Table 2 reports parameter count, operator-specific parameters, multiply-accumulate cost, throughput, and peak GPU memory.

The model architecture consists of an encoder, a recurrent module, and a no-skip decoder. The encoder receives the input tensor $x_{1:T} \in \mathbb{R}^{B \times T \times 3 \times 128 \times 128}$, permuted to $(B, 3, T, 128, 128)$ for 3D convolution. Four sequential blocks, each consisting of two $3 \times 3 \times 3$ Conv3D layers with batch normalization and ReLU, followed by MaxPool3D with kernel $(1, 2, 2)$, gradually reduce spatial resolution from 128 to 64 to 32 to 16 to 8, while channel depth increases from 3 to 64 to 128 to 256 to 512. Temporal resolution is preserved throughout. Adaptive average pooling to $(7, 4, 4)$ followed by reshaping yields the sequence $(B, 7, 8192)$ that feeds the LSTM, since $512 \times 4 \times 4 = 8192$. The recurrent module is a single-layer LSTM operating in k dimensions. At each of the seven time steps, the LSTM-Cell produces a hidden state h_t from the current encoder feature e_t and the previous hidden and cell states. The decoder receives $h_7 \in \mathbb{R}^k$, projects it to $\mathbb{R}^{8192}$ through a linear layer with ReLU activation, reshapes the result to $(B, 512, 4, 4)$, and applies four upsampling blocks that recover spatial resolution through 8, 16, 32, and 64 pixels before a final layer produces the predicted frame at 128×128 with activation tanh. Skip connections are not used between the encoder and decoder. This ensures that everything the decoder knows about the input comes via h_7 .

All models are trained with AdamW using $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay 10^{-5} . The learning rate follows a cosine annealing schedule from $\eta_0 = 2 \times 10^{-5}$ to $\eta_{\min} = 10^{-6}$ over 30 epochs. Gradient norms are clipped to 1.0, and automatic mixed precision is used throughout. The batch size is 8 in all reported runs, with an 80/20 train/validation split using seed 42. The best checkpoint is selected by PSNR valida-

tion and all reported test results come from that checkpoint, evaluated on held-out data without any fine-tuning.

Table 2. Inference efficiency of the five model variants at $k = 64$. FPS is measured at batch size 1 on the evaluation GPU. GMACs are reported per forward pass. Op. params denotes parameters added by the operator block only; the encoder–LSTM–decoder backbone is shared across all variants.

Dataset	Operator	Total Params	Op. Params	GMACs	FPS	GPU (MB)
Indoor	Baseline	20,635,395	0	182.15	65	314
	Householder	20,635,459	64	182.15	65	395
	Givens	20,635,396	1	182.15	65	396
	Composition	20,635,460	65	182.15	65	395
	Lie Algebra	20,637,411	2016	182.15	65	476
KITTI	Baseline	20,635,395	0	182.15	65	396
	Householder	20,635,459	64	182.15	65	395
	Givens	20,635,396	1	182.15	65	476
	Composition	20,635,460	65	182.15	65	397
	Lie Algebra	20,637,411	2016	182.15	65	397
Flying2D	Baseline	20,635,395	0	182.15	65	479
	Householder	20,635,459	64	182.15	63	397
	Givens	20,635,396	1	182.15	63	397
	Composition	20,635,460	65	182.15	65	478
	Lie Algebra	20,637,411	2016	182.15	58	397
Moving3D	Baseline	18,663,171	0	45.54	201	221
	Householder	18,663,235	64	45.54	205	294
	Givens	18,663,172	1	45.54	222	222
	Composition	18,663,236	65	45.54	220	222
	Lie Algebra	18,665,187	2016	45.54	204	295

The complete computational procedure is summarized in Algorithms 1–3. Algorithm 1 describes the encoder–recurrent–decoder forward pass, Algorithm 2 defines the fixed-point operators and their auxiliary losses, and Algorithm 3 gives the training loop used for all model variants.

Algorithm 1 Encoder–Recurrent–Decoder Forward Pass

Require: Input sequence $x_{1:T} \in \mathbb{R}^{B \times T \times 3 \times 128 \times 128}$, encoder E_θ , LSTMCell, decoder D_ϕ , latent dim k

Ensure: Predicted frame $\hat{x}_{T+1} \in \mathbb{R}^{B \times 3 \times 128 \times 128}$

- 1: $\tilde{x} \leftarrow \text{permute}(x_{1:T}, (B, 3, T, 128, 128))$ ▷ reorder for 3D convolution
- 2: $e \leftarrow E_\theta(\tilde{x})$ ▷ 4-block Conv3D encoder; $e \in \mathbb{R}^{B \times 512 \times T \times 8 \times 8}$
- 3: $e \leftarrow \text{AdaptiveAvgPool3D}(e, (7, 4, 4))$ ▷ pool to 7 temporal steps
- 4: $e_{1:7} \leftarrow \text{reshape}(e, (B, 7, 8192))$ ▷ $512 \times 4 \times 4 = 8192$
- 5: $h_0, c_0 \leftarrow \mathbf{0}_k, \mathbf{0}_k$
- 6: **for** $t = 1$ **to** 7 **do**
- 7: $h_t, c_t \leftarrow \text{LSTMCell}(e_t, h_{t-1}, c_{t-1})$
- 8: **end for** ▷ h_7 is the final recurrent hidden state; operator block acts on h_7 during training only
- 9: $z \leftarrow \text{ReLU}(W_{\text{fc}} h_7 + b_{\text{fc}})$ ▷ project $\mathbb{R}^k \rightarrow \mathbb{R}^{8192}$
- 10: $z \leftarrow \text{reshape}(z, (B, 512, 4, 4))$
- 11: $\hat{x}_{T+1} \leftarrow D_\phi(z)$ ▷ 4-block upsample decoder; no skip connections
- 12: **return** $\hat{x}_{T+1}, h_7$

We use $m = 4$ as a common finite-closure diagnostic for all rotation-based operators, not as an assumption that the data possess exact four-fold symmetry.

Algorithm 2 Fixed-Point Operators and Auxiliary Losses

Require: Hidden state $h_7 \in \mathbb{R}^{B \times k}$, operator type $\mathcal{S} \in \{\text{Householder, Givens, Composition, Lie}\}$
Ensure: Auxiliary loss $\mathcal{L}_{\text{fp}}$

- 1: **if** $\mathcal{S} = \text{Householder}$ **then**
- 2: $n \leftarrow \tilde{n} / \|\tilde{n}\|$ ▷ normalise learnable vector $\tilde{n} \in \mathbb{R}^k$, k params
- 3: $\mathcal{M}(h_7) \leftarrow h_7 - 2(n^\top h_7)n$ ▷ reflection through hyperplane $\perp n$
- 4: $\mathcal{L}_{\text{fp}} \leftarrow \|\mathcal{M}(h_7) - h_7\|_1$ ▷ fixed-point loss; pushes h_7 toward $\{h : n^\top h = 0\}$
- 5: **else if** $\mathcal{S} = \text{Givens}$ **then**
- 6: $A \leftarrow (Q - Q^\top) / \|Q - Q^\top\|_F$, Q fixed random ▷ fixed skew-symmetric generator, $A = -A^\top$
- 7: $R(\theta) \leftarrow \exp(\theta A)$ ▷ θ learnable scalar, 1 param; $R \in \text{SO}(k)$
- 8: $h_7^{(0)} \leftarrow h_7$
- 9: **for** $m = 1$ **to** 4 **do**
- 10: $h_7^{(m)} \leftarrow h_7^{(m-1)} R(\theta)^\top$
- 11: **end for**
- 12: $\mathcal{L}_{\text{fp}} \leftarrow \|h_7^{(4)} - h_7\|_1$ ▷ finite closure diagnostic: encourage $R^4(h_7) \approx h_7$
- 13: **else if** $\mathcal{S} = \text{Composition}$ **then**
- 14: *(instantiate $\mathcal{M}$ and $\mathcal{R}$ as above)*
- 15: $\mathcal{L}_{\text{MR}} \leftarrow \|\mathcal{M}(\mathcal{R}(h_7)) - h_7\|_1$ ▷ rotate then reflect
- 16: $\mathcal{L}_{\text{RM}} \leftarrow \|\mathcal{R}(\mathcal{M}(h_7)) - h_7\|_1$ ▷ reflect then rotate
- 17: $\mathcal{L}_{\text{comm}} \leftarrow \|\mathcal{M}(\mathcal{R}(h_7)) - \mathcal{R}(\mathcal{M}(h_7))\|_1$ ▷ commutativity penalty
- 18: $\mathcal{L}_{\text{fp}} \leftarrow \mathcal{L}_{\text{MR}} + \mathcal{L}_{\text{RM}} + \lambda_{\text{comm}} \mathcal{L}_{\text{comm}}$
- 19: **else if** $\mathcal{S} = \text{Lie Algebra}$ **then**
- 20: $\Omega_{ij} \leftarrow +\theta_{ij}$ for $i > j$; $\Omega_{ij} \leftarrow -\theta_{ij}$ for $i < j$; $\Omega_{ii} \leftarrow 0$ ▷ $\frac{k(k-1)}{2}$ learnable params; $\Omega \in \mathfrak{so}(k)$
- 21: $R_\Omega \leftarrow \exp(\Omega)$ ▷ $R_\Omega \in \text{SO}(k)$; full Lie algebra rotation
- 22: $h_7^{(0)} \leftarrow h_7$
- 23: **for** $m = 1$ **to** 4 **do**
- 24: $h_7^{(m)} \leftarrow h_7^{(m-1)} R_\Omega^\top$
- 25: **end for**
- 26: $\mathcal{L}_{\text{fp}} \leftarrow \|h_7^{(4)} - h_7\|_1$ ▷ C4 closure loss
- 27: **end if**
- 28: **return** $\mathcal{L}_{\text{fp}}$

Algorithm 3 Fixed-Point Regularization Training

Require: Dataset $\mathcal{D}$, operator $\mathcal{S}$, epochs $N = 30$, batch size $B = 8$, $k = 64$, $\eta_0 = 2 \times 10^{-5}$, $\eta_{\min} = 10^{-6}$, $\lambda_{\text{fp}} = 1.0$, gradient clip = 1.0, train/val split 80/20 at seed 42
Ensure: Best checkpoint selected by validation PSNR

- 1: Initialise model parameters θ, ϕ ; initialise AdamW optimiser
- 2: Split $\mathcal{D}$ into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$
- 3: **for** epoch $e = 1$ **to** N **do**
- 4: $\eta_e \leftarrow \eta_{\min} + \frac{1}{2}(\eta_0 - \eta_{\min})(1 + \cos(\pi e/N))$ ▷ cosine annealing
- 5: **for** each mini-batch $(x_{1:T}, x_{T+1}) \in \mathcal{D}_{\text{train}}$ **do**
- 6: $\hat{x}_{T+1}, h_7 \leftarrow \text{Forward}(x_{1:T})$ ▷ Algorithm 1
- 7: $\mathcal{L}_{\text{recon}} \leftarrow \|\hat{x}_{T+1} - x_{T+1}\|_1$
- 8: **if** $\mathcal{S} \neq \text{Baseline}$ **then**
- 9: $\mathcal{L}_{\text{fp}} \leftarrow \text{AuxLoss}(h_7, \mathcal{S})$ ▷ Algorithm 2
- 10: $\mathcal{L} \leftarrow \mathcal{L}_{\text{recon}} + \lambda_{\text{fp}} \mathcal{L}_{\text{fp}}$
- 11: **else**
- 12: $\mathcal{L} \leftarrow \mathcal{L}_{\text{recon}}$
- 13: **end if**
- 14: Backpropagate $\mathcal{L}$; clip gradients; update parameters ▷ AMP + gradient clipping to 1.0
- 15: **end for**
- 16: PSNR_{val} $\leftarrow$ Evaluate on $\mathcal{D}_{\text{val}}$ ▷ operator not applied at inference
- 17: **if** PSNR_{val} $>$ PSNR_{best} + δ **then**
- 18: Save checkpoint; PSNR_{best} $\leftarrow$ PSNR_{val}
- 19: **end if**
- 20: **end for**
- 21: **return** best checkpoint

3.2. Fixed-Point Operators

The fixed-point regularizer is instantiated using four algebraic operators acting on the final recurrent hidden state $h_7 \in \mathbb{R}^k$: GlobalHouseholder, GlobalGivens, Composition, and Lie Algebra. Table 3 summarizes their main algebraic properties alongside the unconstrained baseline.

Table 3. Algebraic properties of the geometric operators at $k = 64$. Householder, Givens, and Lie Algebra are norm-preserving transformations; the Composition operator is a symmetrized combination of reflection and rotation terms.

Operator	Group	Params	Algebraic Property	Inverse Used
Baseline	$\{I\}$	0	Identity	I
Householder	$O(k)$	k	$\mathcal{M}^2 = I$	$\mathcal{M}$
Givens	$SO(k)$	1	$\mathcal{R}^{-1} = \mathcal{R}(-\theta)$	$\expm(-\theta A)$
Composition	Symmetrized maps	$k + 1$	$\frac{1}{2}(\mathcal{M}\mathcal{R} + \mathcal{R}\mathcal{M})$	No
Lie Algebra	$SO(k)$	2016	Full $SO(k)$ rotation	$\expm(-\Omega)$

The operators differ not only in parameter count but also in how strongly they can restrict h_7 . GlobalGivens uses a single learnable scalar angle with a fixed skew-symmetric generator, so its four-step closure loss encourages rotational consistency while leaving most latent directions comparatively unconstrained. Lie Algebra instead learns a full $so(k)$ generator with $k(k - 1)/2$ parameters, equal to 2016 at $k = 64$, and can therefore constrain many latent planes simultaneously. This stronger parameterization can become a stronger restriction under the fixed-point objective, which motivates the over-constraint analysis in Sections 4 and 5.

3.3. Training Datasets

Four datasets are used to cover a range of scene structures: (i) Indoor Buildings, (ii) KITTI, (iii) Flying Shapes 2D, and (iv) Moving 3D Shapes. Their frame counts, sequence counts, and train–test split structure are summarized in Table 4. The datasets are explained below:

- The Indoor Buildings dataset consists of RGB video recorded with an Intel RealSense D435 depth camera in university buildings. Scenes include elevator interiors, lobbies, corridors, and doorway areas: spaces that are architecturally structured, with walls, door frames, floor boundaries, and signage that frequently exhibit approximate bilateral symmetry. The training pool contains 7213 frames from five indoor folders, resized to 128×128 and normalized to $[-1, 1]$. Training and validation are obtained from this training pool using an 80/20 split with seed 42. The held-out Indoor evaluation set is collected from different buildings/environments and is kept completely separate from training and validation; it contains 12,978 frames, corresponding to 12,852 input-target sequences.
- The KITTI dataset [41] provides RGB driving video recorded from a moving vehicle in urban and suburban environments in Karlsruhe, Germany. Scenes include roads, parked vehicles, buildings, pedestrians, and vegetation. Unlike the Indoor dataset, KITTI scenes have weaker and less consistent bilateral regularity. The road layout is often approximately symmetric about the vehicle axis, but other scene elements are not, and camera motion introduces continuous viewpoint change. Frames are resized to 128×128 . Training uses 29 drives and evaluation uses five completely held-out drives under a drive-disjoint protocol, giving $n = 1196$ test sequences.
- The Flying Shapes 2D dataset consists of synthetic RGB sequences of simple 2D geometric shapes moving across a uniform background. The shapes and background

produce strong geometric regularity relative to natural video, making this dataset a controlled setting for testing whether algebraic operators interact with simple spatial structure. The training split contains 10,000 sequences (190,000 frames), and the test split contains 2000 sequences (38,000 frames).

- The Moving 3D Shapes dataset consists of synthetic sequences of 3D geometric objects rendered under varying viewpoints. Unlike Flying Shapes 2D, viewpoint rotation introduces perspective distortion and shading variation that break the exact pixel-space regularity present in the 2D case. The dataset is therefore a harder test for the geometric operators: the objects are regular in 3D space, but their projections onto the image plane depend on viewpoint and lighting. The training split contains 10,000 sequences (190,000 frames), and the test split contains 2000 sequences (38,000 frames).

Table 4. Dataset composition used in the experiments. Training and validation are split from the training pool using an 80/20 split with seed 42. Test data are held out separately. Sequence counts assume a sliding window of $T + 1 = 19$ frames with stride 1 for Indoor and KITTI and predefined 19-frame sequences for the synthetic datasets.

Dataset	Train Frames	Train Seq.	Test Frames	Test Seq.
Indoor Buildings	7213	7123	12,978	12,852
KITTI	6785	6155	1286	1196
Flying Shapes 2D	190,000	10,000	38,000	2000
Moving 3D Shapes	190,000	10,000	38,000	2000

The baseline and four fixed-point variants are trained under a fixed seed and identical protocol, including the same data split, initialization protocol, optimizer, schedule, architecture, and evaluation pipeline. The ConvLSTM-cell is evaluated separately as a controlled recurrent-cell ablation. The results are interpreted as controlled comparative evidence for isolating operator-induced latent reorganization rather than as an estimate of variability over random initializations. The paired Wilcoxon tests and the bootstrap intervals for entropy and activation variance assess consistency over the held-out evaluation population, and they do not replace independent multi-seed replication.

3.4. Evaluation Metrics

In this paper, we used four latent structure metrics to evaluate the latent reorganization induced by each operator. The latent structure metrics are:

- Shannon entropy H measures how evenly squared activation energy is distributed across latent dimensions. For each hidden state, squared activations are normalized across dimensions, averaged over all test sequences, and converted to entropy using base-2 logarithms.
- Dims@90% is computed from the centered hidden-state matrix $H_7 - \bar{H}_7$. It counts the number of singular-value energy components needed to explain 90% of the total squared singular-value energy. Larger values indicate broader sample-level latent variation.
- Effective rank is computed from the centered matrix $H_7 - \bar{H}_7$. Let $\{s_i\}$ be its singular values and $p_i = s_i / \sum_j s_j$. The effective rank is $\exp(-\sum_i p_i \log p_i)$.
- Activation variance is the mean per-dimension variance of h_7 across all N test sequences, computed as the average diagonal of the sample covariance matrix of $H_7 \in \mathbb{R}^{N \times k}$.

These metrics do not demonstrate semantic disentanglement or mutual information compression. They are used only as diagnostic summaries of how activation energy and sample-level variation are distributed throughout the recurrent state.

4. Results and Discussion

In our experiments, we first examine whether fixed-point regularization changes the recurrent latent state by compressing it or reorganizing how its dimensions are used. The relevant object is the final LSTM hidden state $h_7 \in \mathbb{R}^k$, since this is the only representation passed from the encoder-recurrent module to the decoder. We characterize the hidden state via Shannon entropy, Dims@90, effective rank, and activation variance.

4.1. Reconstruction Quality

Reconstruction quality is measured using four complementary metrics. The Peak Signal-to-Noise Ratio (PSNR) is the ratio of the maximum possible pixel intensity to the mean-squared prediction error, expressed in decibels. Higher values indicate better pixel-level fidelity. The structural similarity index (SSIM) measures the perceptual agreement between the predicted and ground-truth frames in terms of luminance, contrast, and structure; higher values are better. The Learned Perceptual Image Patch Similarity (LPIPS) [42] measures the perceptual distance using deep feature representations extracted from a pretrained AlexNet. Lower values indicate a higher perceptual similarity. Latent metrics explain how the hidden state changes, while MAE/PSNR/SSIM/LPIPS show the task-level effect. Reconstruction quality metrics are shown in Table 5. The ConvLSTM-cell row is included as a controlled recurrent-cell reference under the same bottleneck interface and matched evaluation convention.

Table 5. Reconstruction quality on held-out splits. For PSNR, values in parentheses denote absolute gains or losses in dB relative to the baseline. For SSIM, LPIPS, and MAE, values denote percentage change relative to the baseline. Bold indicates the best value for each metric within each dataset. $\uparrow$ higher is better; $\downarrow$ lower is better.

Dataset	Model	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	MAE $\downarrow$
Indoor	Baseline	17.349	0.551	0.687	0.1087
	ConvLSTM-cell	17.552 (+0.20 dB)	0.554 (+0.56%)	0.655 (−4.67%)	0.1070 (−1.62%)
	Householder	18.173 (+0.82 dB)	0.571 (+3.62%)	0.639 (−6.99%)	0.0985 (−9.38%)
	Givens	18.390 (+1.04 dB)	0.576 (+4.54%)	0.660 (−3.93%)	0.0964 (−11.31%)
	Composition	18.143 (+0.79 dB)	0.564 (+2.40%)	0.649 (−5.53%)	0.0986 (−9.29%)
	Lie Algebra	18.143 (+0.79 dB)	0.571 (+3.62%)	0.640 (−6.84%)	0.0992 (−8.74%)
KITTI	Baseline	14.580	0.389	0.631	0.1194
	ConvLSTM-cell	13.959 (−0.62 dB)	0.372 (−4.27%)	0.662 (+4.91%)	0.1315 (+10.13%)
	Householder	14.784 (+0.20 dB)	0.395 (+1.39%)	0.637 (+0.86%)	0.1155 (−3.27%)
	Givens	14.682 (+0.10 dB)	0.394 (+1.13%)	0.632 (+0.18%)	0.1170 (−2.01%)
	Composition	14.570 (−0.01 dB)	0.388 (−0.34%)	0.620 (−1.74%)	0.1205 (+0.86%)
	Lie Algebra	14.698 (+0.12 dB)	0.398 (+2.33%)	0.637 (+0.92%)	0.1182 (−1.08%)
Flying2D	Baseline	21.387	0.948	0.191	0.0130
	ConvLSTM-cell	21.811 (+0.42 dB)	0.948 (−0.02%)	0.180 (−5.76%)	0.0133 (+2.54%)
	Householder	21.718 (+0.33 dB)	0.950 (+0.26%)	0.181 (−5.76%)	0.0118 (−9.52%)
	Givens	21.405 (+0.02 dB)	0.949 (+0.16%)	0.181 (−5.24%)	0.0129 (−0.77%)
	Composition	21.401 (+0.01 dB)	0.950 (+0.19%)	0.183 (−4.19%)	0.0127 (−2.31%)
	Lie Algebra	21.386	0.950 (+0.25%)	0.187 (−2.36%)	0.0125 (−3.85%)
Moving3D	Baseline	16.190	0.712	0.140	0.0678
	ConvLSTM-cell	16.494 (+0.30 dB)	0.707 (−0.70%)	0.146 (+4.29%)	0.0695 (+2.51%)
	Householder	16.110 (−0.08 dB)	0.710 (−0.21%)	0.141 (+0.21%)	0.0682 (+0.53%)
	Givens	16.147 (−0.04 dB)	0.712 (+0.01%)	0.140 (−0.36%)	0.0677 (−0.15%)
	Composition	15.960 (−0.23 dB)	0.706 (−0.80%)	0.140 (−0.36%)	0.0702 (+3.51%)
	Lie Algebra	15.952 (−0.24 dB)	0.686 (−3.67%)	0.142 (+1.14%)	0.0721 (+6.34%)

These reconstruction metrics are used as task-level anchors rather than as a claim of state-of-the-art video prediction. The main analysis concerns whether the same prediction task reveals systematic changes in the recurrent latent geometry under different algebraic constraints.

The strongest fixed-point gains are observed on the structured Indoor setting, where GlobalGivens improves PSNR by +1.04 dB, increases SSIM by 4.54%, reduces LPIPS by 3.93%, and reduces MAE by 11.31% relative to the baseline. On KITTI, the gains are more modest, with Householder providing the best PSNR and MAE among the regularized variants. On Moving 3D Shapes, several operators no longer improve reconstruction, indicating that the effectiveness of the algebraic prior depends on how well the imposed latent geometry aligns with the visible scene structure.

To test whether the reconstruction differences in Table 5 are consistent across held-out samples, we performed paired Wilcoxon signed-rank tests between each regularized model and the baseline on matched test sequences. The test was applied separately for each dataset, operator, and metric. These tests quantify whether the per-sequence differences are consistently shifted relative to the baseline, but they do not estimate variability across independently trained random seeds. The results are reported in Table 6.

Table 6. Paired Wilcoxon signed-rank tests between each model and the baseline on the full matched held-out test set. $\uparrow$ higher is better; $\downarrow$ lower is better. Each cell reports the mean paired difference Δ (model – baseline) with a significance marker. For PSNR and SSIM, positive values indicate improvement; for MAE, negative values indicate improvement. The ConvLSTM-cell row is included only as a controlled recurrent-cell reference. Significance: *** $p < 0.001$ and ^{ns} = not significant.

Dataset	Operator	Δ PSNR (dB) $\uparrow$	Δ SSIM $\uparrow$	Δ MAE $\downarrow$
Indoor	ConvLSTM-cell	+0.126 ***	+0.0061 ***	−0.00221 ***
	Householder	+0.672 ***	+0.0195 ***	−0.00930 ***
	Givens	+0.768 ***	+0.0230 ***	−0.01078 ***
	Composition	+0.682 ***	+0.0160 ***	−0.00973 ***
	Lie Algebra	+0.604 ***	+0.0190 ***	−0.00805 ***
KITTI	ConvLSTM-cell	−0.643 ***	−0.0165 ***	+0.01207 ***
	Householder	+0.201 ***	+0.0054 ***	−0.00395 ***
	Givens	+0.099 ***	+0.0045 ***	−0.00239 ***
	Composition	−0.018 ^{ns}	−0.0016 ***	+0.00104 ***
	Lie Algebra	+0.112 ***	+0.0090 ***	−0.00129 ***
Flying2D	ConvLSTM-cell	−0.156 ***	+0.0001 ***	+0.00031 ***
	Householder	+0.170 ***	+0.0025 ***	−0.00124 ***
	Givens	−0.153 ***	+0.0014 ***	−0.00012 ^{ns}
	Composition	−0.207 ***	+0.0018 ***	−0.00032 ***
	Lie Algebra	−0.194 ***	+0.0024 ***	−0.00049 ***
Moving3D	ConvLSTM-cell	−0.199 ***	−0.0049 ***	+0.00169 ***
	Householder	−0.078 ***	−0.0015 ***	+0.00037 ***
	Givens	−0.044 ***	+0.0000 ^{ns}	−0.00010 ^{ns}
	Composition	−0.231 ***	−0.0057 ***	+0.00239 ***
	Lie Algebra	−0.232 ***	−0.0262 ***	+0.00431 ***

The paired tests show that the effect of fixed-point regularization is dataset- and operator-dependent. On Indoor Buildings, all four operators significantly improve PSNR, SSIM, and MAE over the baseline. On KITTI, Householder, Givens, and Lie Algebra improve the distortion metrics, whereas Composition is mixed. On Flying Shapes 2D, Householder gives the clearest improvement, while the other operators improve SSIM and MAE but reduce PSNR. On Moving 3D Shapes, most operators degrade PSNR and MAE, with Givens closest to neutral. This pattern is consistent with the role of scene structure. Indoor and Flying Shapes 2D contain stronger image-plane regularity, including repeated architectural layouts or simple synthetic geometry, and show the clearest benefits from lightweight algebraic constraints. KITTI contains weaker and less consistent regularity because traffic, occlusion, vegetation, illumination changes, and camera motion disrupt

the visible structure. Moving 3D Shapes contains object-level geometric symmetry, but viewpoint-dependent 3D rotations induce changing 2D projections, perspective effects, shading, and self-occlusion; therefore, its pixel-space regularity is less consistent. The ConvLSTM-cell control is interpreted as a recurrent-cell reference under the same bottleneck interface, rather than as a separate high-capacity video-prediction benchmark. The mean per-sequence differences in Table 6 reflect the typical sequence-level shift and may differ from the aggregate PSNR differences in Table 5 when a small number of sequences with large gains raise the aggregate mean while the majority of sequences show a marginal loss; this accounts for the sign difference observed for ConvLSTM-cell, Givens, Composition, and Lie Algebra on Flying Shapes 2D.

4.2. Latent Reorganization

We next examine how each operator changes the final recurrent bottleneck state h_7 . The goal is to distinguish useful latent reorganization from simple dimensional compression.

Table 7 shows that the lighter operators generally preserve or broaden active latent participation. Householder, Givens, and Composition maintain or increase Dims@90 on most datasets while avoiding the strong activation-variance collapse seen in Lie Algebra. This indicates that the useful regularizers do not simply compress the latent state; they tend to redistribute latent activity while preserving broad participation. PCA and t-SNE projections for the baseline, four fixed-point variants, and the ConvLSTM-cell control are provided in Appendix A.2 Figure A6 as complementary qualitative views of the latent organization.

Table 7. Latent structure metrics on h_7 computed on the full held-out test set. Percentage change vs. baseline in parentheses. Bold indicates the strongest activation-variance reduction relative to the baseline within each dataset.

Dataset	Model	Entropy	Dims@90	Eff. Rank	Act. Var
Indoor	Baseline	5.261	18	42.93	0.0329
	ConvLSTM-cell	5.593 (+6.32%)	17 (−5.6%)	41.38 (−3.60%)	0.0305 (−7.42%)
	Householder	5.354 (+1.77%)	20 (+11.1%)	43.47 (+1.24%)	0.0311 (−5.43%)
	Givens	5.252 (−0.16%)	19 (+5.6%)	42.59 (−0.79%)	0.0286 (−13.08%)
	Composition	5.381 (+2.28%)	20 (+11.1%)	43.34 (+0.94%)	0.0309 (−6.12%)
	Lie Algebra	5.369 (+2.06%)	17 (−5.6%)	41.79 (−2.67%)	0.0024 (−92.69%)
KITTI	Baseline	5.341	11	33.40	0.0254
	ConvLSTM-cell	5.500 (+2.97%)	8 (−27.3%)	29.78 (−10.83%)	0.0545 (+114.72%)
	Householder	5.423 (+1.53%)	11 (+0.0%)	32.59 (−2.44%)	0.0261 (+3.02%)
	Givens	5.234 (−2.00%)	12 (+9.1%)	33.75 (+1.04%)	0.0202 (−20.44%)
	Composition	5.211 (−2.44%)	11 (+0.0%)	32.47 (−2.81%)	0.0233 (−8.06%)
	Lie Algebra	5.109 (−4.34%)	10 (−9.1%)	30.79 (−7.81%)	0.0016 (−93.57%)
Flying2D	Baseline	5.658	8	28.99	0.0345
	ConvLSTM-cell	5.694 (+0.64%)	10 (+25.0%)	34.87 (+20.28%)	0.0271 (−21.54%)
	Householder	5.557 (−1.78%)	10 (+25.0%)	32.04 (+10.53%)	0.0250 (−27.69%)
	Givens	5.700 (+0.75%)	10 (+25.0%)	31.31 (+8.02%)	0.0347 (+0.43%)
	Composition	5.631 (−0.47%)	10 (+25.0%)	31.92 (+10.10%)	0.0303 (−12.23%)
	Lie Algebra	5.283 (−6.62%)	10 (+25.0%)	29.95 (+3.33%)	0.0014 (−96.06%)
Moving3D	Baseline	5.431	7	27.32	0.0150
	ConvLSTM-cell	5.293 (−2.53%)	7 (+0.0%)	27.12 (−0.75%)	0.0122 (−18.45%)
	Householder	5.291 (−2.57%)	8 (+14.3%)	27.87 (+2.02%)	0.0149 (−0.42%)
	Givens	5.504 (+1.34%)	8 (+14.3%)	28.09 (+2.80%)	0.0190 (+26.55%)
	Composition	5.356 (−1.38%)	7 (+0.0%)	26.12 (−4.38%)	0.0178 (+18.83%)
	Lie Algebra	4.921 (−9.38%)	5 (−28.6%)	18.93 (−30.69%)	0.0028 (−81.50%)

Figure 3 summarizes Dims@90 across the four datasets.

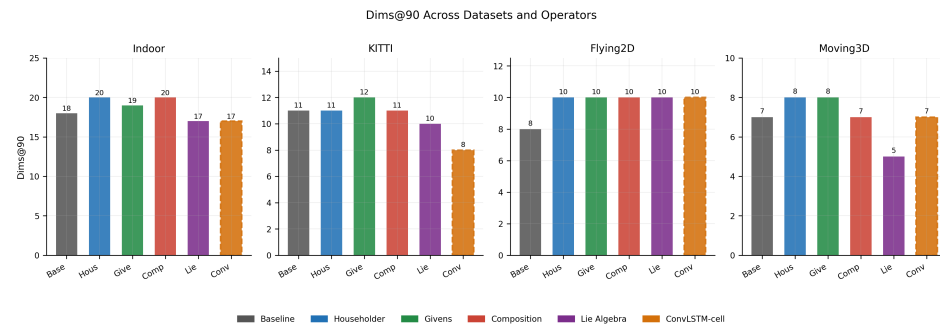


Figure 3. Dims@90 across datasets and operators.

Effective rank and activation variance support the same interpretation. The lighter operators produce measured changes: the latent state becomes less dominated by a few high-variance coordinates but remains broadly active. Figure 4 makes this structure visible. The baseline concentrates energy in a small set of dimensions, while lighter operators distribute energy across a wider part of the hidden space.

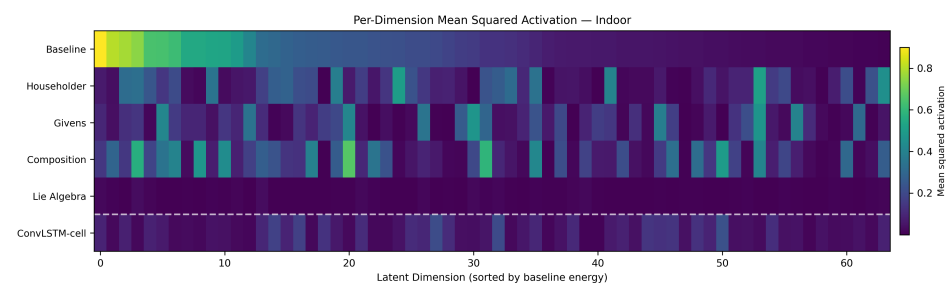


Figure 4. Per-dimension mean squared activation for the Indoor dataset, sorted by baseline activation magnitude. The heatmap shows how each operator redistributes activation energy across latent dimensions relative to the baseline. The dotted horizontal line separates the ConvLSTM-cell control from the baseline and fixed-point variants.

The Lie Algebra operator behaves differently. It suppresses activation variance by more than 80% across all datasets, indicating strong concentration of latent activity. This suggests that the higher-capacity Lie operator can satisfy the fixed-point constraint by suppressing variation across a broad part of h_7 , whereas lighter operators preserve more usable latent degrees of freedom. This concentration is accompanied by lower Dims@90 on Indoor, KITTI, and Moving 3D Shapes but not on Flying Shapes 2D. However, the variance collapse does not translate into the most reliable prediction behavior. This suggests that stronger latent concentration is not necessarily beneficial; in some cases, it can remove useful predictive variation and make the representation more brittle.

As an additional robustness check, we computed nonparametric bootstrap confidence intervals for entropy and activation variance using 2000 resamples of the held-out hidden-state arrays; the results are reported in Appendix A.3 Table A1. This analysis is used only as a supporting diagnostic. The primary latent-geometry results remain the full held-out descriptors reported in Table 7. Entropy separates from the baseline for all operators across the four datasets, indicating that the fixed-point constraints consistently change the distribution of latent energy. Activation variance also separates from the baseline in most cases, with the strongest and most consistent separation observed for Lie Algebra. Thus, the bootstrap intervals provide reassurance that the entropy and activation-variance trends are not driven by a small number of held-out sequences, but they are not used as the main basis for the latent-geometry claims.

The scatter plots test whether the prediction quality follows latent compression. Figure 5 compares MAE against Dims@90. The additional rank and entropy plots are

reported in Appendix A.3. The same pattern appears in all cases: better prediction does not consistently occur at lower dimensionality, lower effective rank, or lower entropy. The best models often use equal or greater latent participation than the baseline, while Lie Algebra can produce the most compressed representation without producing the best prediction. This separates latent reorganization from dimensional compression. Fixed-point regularization helps when the latent state becomes more operator-compatible while remaining flexible. It does not help simply because the representation becomes smaller.

Additional latent geometry diagnostics including cumulative energy profiles and entropy-PSNR scatter plots are provided in Appendix A.2.

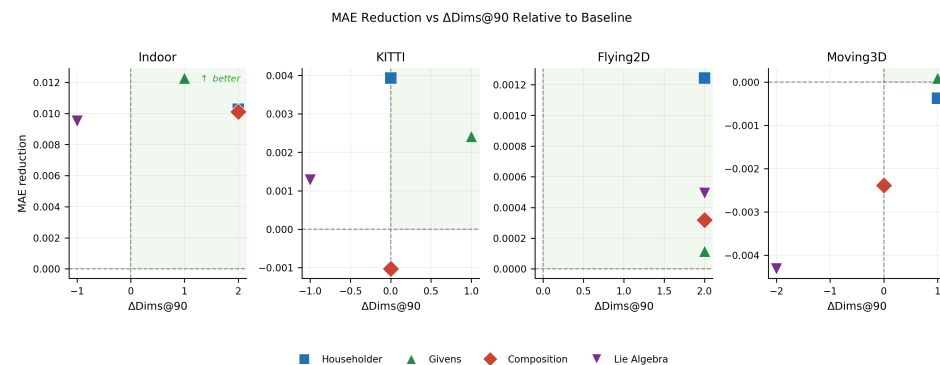


Figure 5. Relationship between MAE and Dims@90 across operators and datasets. Lower MAE indicates better prediction accuracy, while larger Dims@90 indicates broader latent participation. The green-shaded region indicates models that preserve or increase Dims@90 while reducing MAE relative to the baseline. The dashed vertical and horizontal lines indicate zero change relative to the baseline.

4.3. Latent Perturbation Robustness

We then test whether the learned latent organization remains stable under the imposed algebraic constraint and under direct perturbation of h_7 . These diagnostics help separate robust reorganization from brittle over-concentration.

The closure-loss and noise experiments provide diagnostic support for this interpretation. Givens and Composition satisfy their algebraic constraints almost exactly, while the Householder retains only a small residual. Lie Algebra satisfies its constraint on the easier datasets but becomes unstable on Moving 3D Shapes, where its prediction behavior is also weakest. Full closure-loss figures are reported in Appendix A.3.

The same failure mode appears under latent Gaussian noise. The lighter operators degrade mildly, whereas the Lie Algebra degrades much more strongly. Under perturbation, its compact, clean representation expands sharply across latent dimensions, showing that aggressive concentration has made the representation brittle rather than robust. Figure 6 reports this degradation directly. The corresponding Dims@90, effective rank, and closure-loss behavior under noise are reported in Appendix A.3.

Table 8 consolidates this behavior within the noise-sweep evaluation pass: the number of active dimensions before and after perturbation, the resulting effective rank, and the MAE degradation for every model and dataset.

4.4. Cross-Dataset Consistency and Interpretation

Finally, we summarize the directional behavior of the operators across datasets and relate these patterns to the main claim. This comparison identifies which operators support latent reorganization and which reveal over-constraint failure modes.

Table 9 summarizes the directional behavior of the operators across datasets. Householder, Givens, and Composition generally preserve or broaden latent participation while

remaining comparatively stable under perturbation. In contrast, Lie Algebra produces the strongest activation-variance collapse and is also the most sensitive to latent noise. This indicates that the most reliable fixed-point regularizers are not simply those that compress the latent state most strongly but those that reorganize the latent representation while preserving sufficient active capacity.

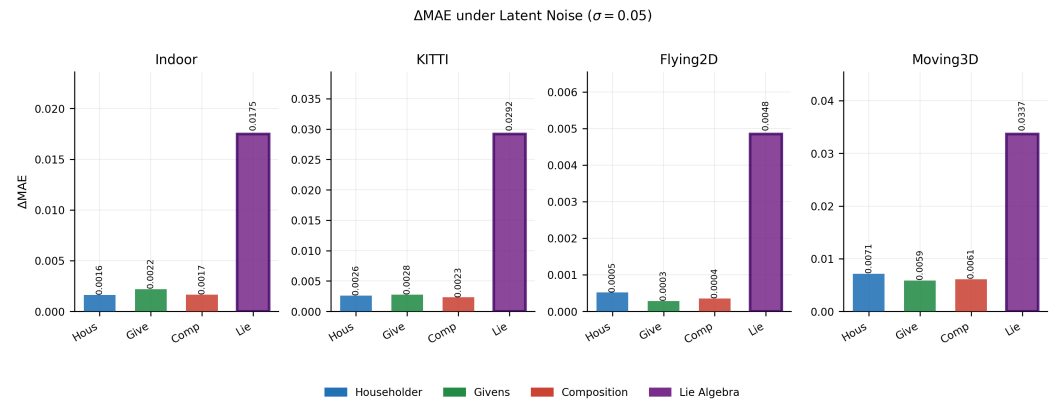


Figure 6. Δ MAE under latent Gaussian noise ($\sigma = 0.05$) for the regularized operators. Values are computed within the separate noise-sweep evaluation pass.

Table 8. Latent dimensionality and reconstruction degradation under inference-time Gaussian noise ($\sigma = 0.05$ injected into h_7). Dims@90 and effective rank are computed within the separate noise-sweep evaluation pass. The $\sigma = 0$ values in this table therefore serve as the clean baseline for the perturbation sweep and are not intended to exactly duplicate the clean latent-summary values in Table 7. Δ MAE = $\text{MAE}(\sigma = 0.05) - \text{MAE}(\sigma = 0)$. Bold indicates the largest noisy Dims@90, effective rank, and Δ MAE value within each dataset.

Dataset	Model	Clean Dims@90	Noisy Dims@90	Eff. Rank ($\sigma = 0.05$)	Δ MAE
Indoor	Baseline	11	18	44.27	0.0015
	Householder	12	19	44.66	0.0016
	Givens	11	19	44.76	0.0022
	Composition	11	19	44.97	0.0017
	Lie Algebra	11	51	60.36	0.0175
KITTI	Baseline	11	19	44.27	0.0020
	Householder	11	18	43.95	0.0026
	Givens	12	22	45.96	0.0028
	Composition	11	19	44.68	0.0023
	Lie Algebra	10	51	59.77	0.0292
Flying2D	Baseline	8	12	40.41	0.0003
	Householder	10	18	44.11	0.0005
	Givens	10	14	41.62	0.0003
	Composition	10	16	42.81	0.0004
	Lie Algebra	10	52	60.77	0.0048
Moving3D	Baseline	7	23	45.52	0.0066
	Householder	8	24	46.21	0.0071
	Givens	8	19	44.16	0.0059
	Composition	7	19	44.02	0.0061
	Lie Algebra	5	48	56.52	0.0337

This distinction is important because compression and generalization are not equivalent in these experiments. Lie Algebra can improve over the baseline on some datasets, but its aggressive variance suppression does not consistently produce the best prediction behavior and degrades most clearly on Moving 3D Shapes. The stronger parameterization of the Lie operator can satisfy the fixed-point criterion by limiting variation over a wider

portion of h_7 , which becomes a drawback when the imposed geometry is poorly matched to the data. The lighter operators instead appear to encourage a more coherent use of latent dimensions without forcing the representation into an overly concentrated form. We therefore interpret the observed effect as structural latent reorganization rather than classical dimensional compression.

Table 9. Directional consistency of latent-geometry behavior across datasets. $\uparrow$ indicates an increase relative to the baseline, while $\downarrow$ indicates a decrease relative to the baseline. Counts indicate the number of datasets (out of 4), showing the stated direction relative to the baseline. $^+$ Lie Algebra achieves the strongest activation variance reduction, but this reflects over-constraint rather than reorganization (see Section 4).

Model	Dims@90 $\uparrow$	Eff. Rank $\uparrow$	Act. Var $\downarrow$	Noise Sensitivity
Householder	increased (3)/ preserved (1)	increased (3/4)	decreased (3/4)	low
Givens	increased (4/4)	increased (3/4)	mixed (2 $\uparrow$ /2 $\downarrow$)	low
Composition	increased (2)/ preserved (2)	mixed (2 $\uparrow$ /2 $\downarrow$)	decreased (3/4)	low
Lie Algebra	decreased (3/4)	decreased (3/4)	decreased (4/4) $^+$	high

The dataset dependence further supports this view. Indoor and Flying Shapes 2D contain stronger image-plane regularity, such as repeated architectural structure or simple synthetic geometry, and therefore show clearer gains from the lightweight fixed-point operators. KITTI contains weaker and less consistent regularity due to traffic, occlusion, illumination changes, and camera motion. Moving 3D Shapes contains object-level geometric symmetry, but viewpoint-dependent 3D rotation changes the 2D projection through perspective, shading, and self-occlusion. Thus, the algebraic prior behaves as a geometry-sensitive inductive bias rather than a universal regularizer.

A direct comparison with high-capacity video predictors such as ConvLSTM [23], PredRNN [24], PredRNN++ [25], or Eidetic 3D LSTM [26] is architecturally inappropriate for the identification goal of this study. These models maintain spatiotemporal hidden states or memory tensors, typically $h \in \mathbb{R}^{C \times H \times W}$, whereas the fixed-point operators studied here act on a flat recurrent vector $h_7 \in \mathbb{R}^k$. The geometric constraint is defined directly on $\mathbb{R}^k$, so applying the same operator to convolutional hidden tensors would change the object of study. Furthermore, high-capacity video predictors often route information to the decoder through multiple pathways, including spatial skip connections, gradient highway units, attention or memory modules, and convolutional transitions. In such architectures, the decoder is not solely dependent on the final recurrent state, making it difficult to attribute changes in performance to the organization of a single latent bottleneck. The no-skip flat-bottleneck architecture is therefore a deliberate identification design rather than a capacity-driven baseline choice. As a controlled compromise, we include a ConvLSTM-cell control as a lightweight surrogate reference for spatiotemporal recurrent modeling. It keeps the same encoder, no-skip decoder, $k = 64$ bottleneck interface, optimizer, and evaluation protocol, while replacing only the recurrent transition.

Finally, the theoretical argument is supported by latent proxy metrics rather than direct mutual information estimation, so the results should be read as empirical evidence for latent reorganization rather than a complete mathematical proof.

5. Conclusions

This paper introduced fixed-point regularization of LSTM hidden states as a training-time geometric prior for recurrent frame prediction. Across four held-out datasets and

four algebraic operators, the results show that improved prediction is not explained by aggressive latent compression alone. The operators that hold up best are those that preserve or broaden active latent participation while reorganizing how the hidden dimensions are used.

Among the tested operators, Householder and Givens provide the clearest support for this interpretation. Householder gives the most balanced evidence of latent reorganization, combining stable prediction gains with broader or preserved latent participation and low perturbation sensitivity. Givens shows the strongest expansion of active latent dimensions and the most consistent task-level behavior across datasets, especially when consistency is understood as avoiding large degradation while maintaining broad latent use. Composition follows the same general direction but less consistently. Together, these lighter operators support the main claim: useful fixed-point regularization acts less like a dimensional bottleneck and more like a geometric bias that organizes the recurrent state into a more operator-compatible form.

Lie Algebra provides an important contrast. It produces the strongest activation-variance collapse, reducing activation variance by approximately 81–96% across datasets, but this concentration does not translate into the most reliable reconstruction behavior and is accompanied by high sensitivity to latent perturbation. This identifies an over-constraint failure mode: when the imposed operator is too strong or poorly aligned with the data geometry, the latent state can become overly concentrated, discard useful predictive variation, and degrade under noise.

These findings suggest that lightweight algebraic constraints can be useful training-time priors for organizing recurrent latent dynamics, particularly when the scene structure is compatible with the imposed geometry. At the same time, the results show that fixed-point regularization is not a universal improvement mechanism. Its benefit depends on the match between the algebraic prior, the latent bottleneck, and the visual structure of the data.

The present experiments use a fixed-seed controlled protocol to isolate operator effects under matched training conditions. The paired tests and bootstrap intervals support consistency over held-out evaluation samples, but they do not quantify training-run variability across independent random initializations. Reporting mean and standard deviation over multi-seed training runs remains an important direction for broader empirical validation.

Beyond the present no-skip bottleneck setting, recent work on Mamba-based defect segmentation and multi-task thermal biometric learning illustrates complementary directions for richer visual architectures, including state-space feature extraction and multi-task supervision [43,44]. We therefore identify the extension of fixed-point latent regularization to state-space and multi-task architectures as future work, while keeping the present study focused on isolating latent-space reorganization under a strict recurrent bottleneck.

Author Contributions: Conceptualization, A.G.; methodology, A.G.; software, A.G.; validation, A.G.; formal analysis, A.G.; investigation, A.G.; data curation, A.G. and J.K.; writing original draft preparation, A.G.; writing review and editing, A.G., J.K., Y.-C.A. and T.-Y.K.; visualization, A.G.; supervision, T.-Y.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a grant agreement with the Korea Institute for Advancement of Technology (KIAT), project number P0028922.

Data Availability Statement: The public KITTI dataset is available from its original source [41]. Our trained model checkpoints and training scripts are publicly available on Zenodo (<https://doi.org/10.5281/zenodo.20936843>). The indoor building video data are available upon reasonable request subject to institutional and privacy constraints.

Conflicts of Interest: The authors declare no conflicts of interest. All authors have read and agreed to the published version of the manuscript.

Appendix A. Additional Diagnostics

Appendix A.1. Training and Convergence

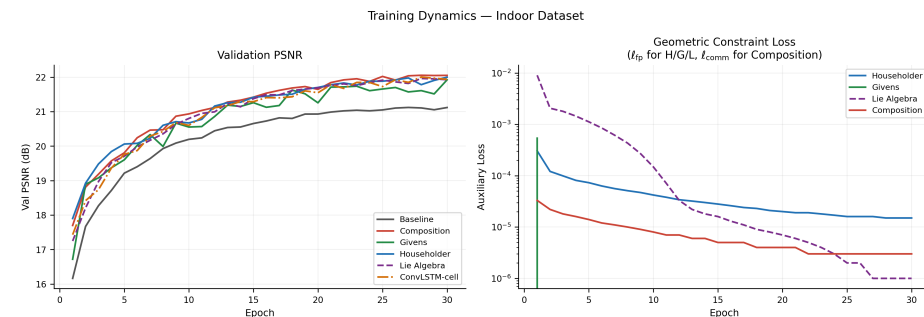


Figure A1. Training dynamics on the Indoor dataset. **Left:** validation PSNR curves for the baseline and four fixed-point variants over 30 epochs. **Right:** auxiliary geometric constraint losses on a log scale. The figure is included to verify convergence behavior rather than to serve as a reconstruction-quality comparison.

Householder shows smooth auxiliary-loss convergence, while Givens rapidly stabilizes near the loss floor. Lie Algebra starts from a larger constraint loss and decreases over a longer portion of the training schedule, consistent with its stronger $so(k)$ parameterization.

Appendix A.2. Additional Latent Geometry Diagnostics

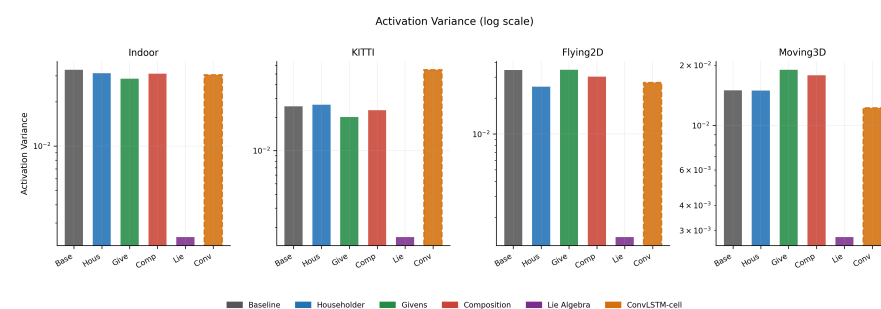


Figure A2. Activation variance of the final recurrent hidden state h_7 on a log scale across all four datasets for the baseline, four fixed-point variants, and the ConvLSTM-cell control.

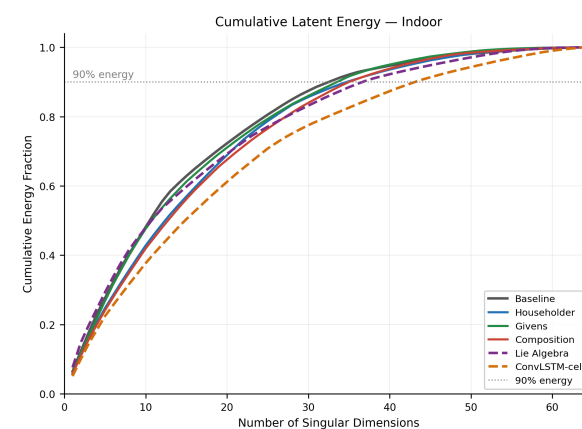


Figure A3. Cumulative singular-value energy as a function of the number of singular dimensions for the Indoor held-out hidden-state matrix H_7 . The d values in the legend and markers indicate where each model's cumulative energy profile crosses the 90% threshold, consistent with the Dims@90 values in Table 7.

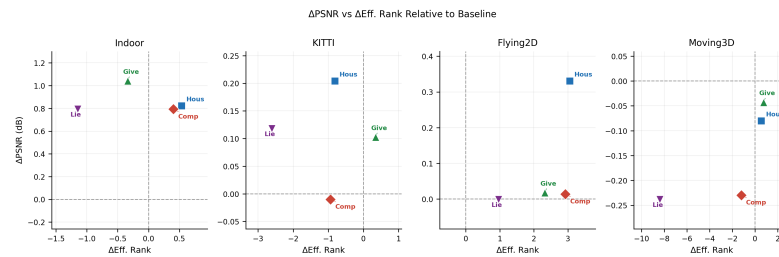


Figure A4. Δ PSNR versus Δ effective rank relative to baseline across all four datasets. Points in the upper-right quadrant indicate operators that improve both PSNR and effective rank simultaneously.

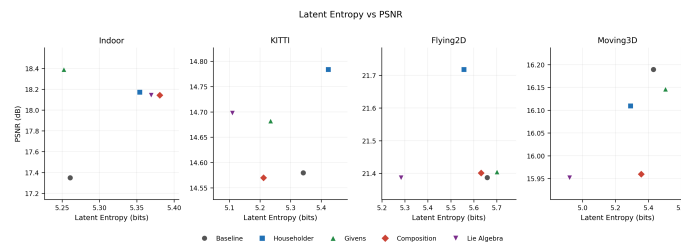


Figure A5. Latent entropy versus PSNR across all four datasets. Each point corresponds to one operator variant.



Figure A6. PCA and t-SNE projections of h_7 on the Indoor test set for the baseline, four fixed-point variants, and the ConvLSTM-cell control.

Appendix A.3. Constraint and Noise Diagnostics

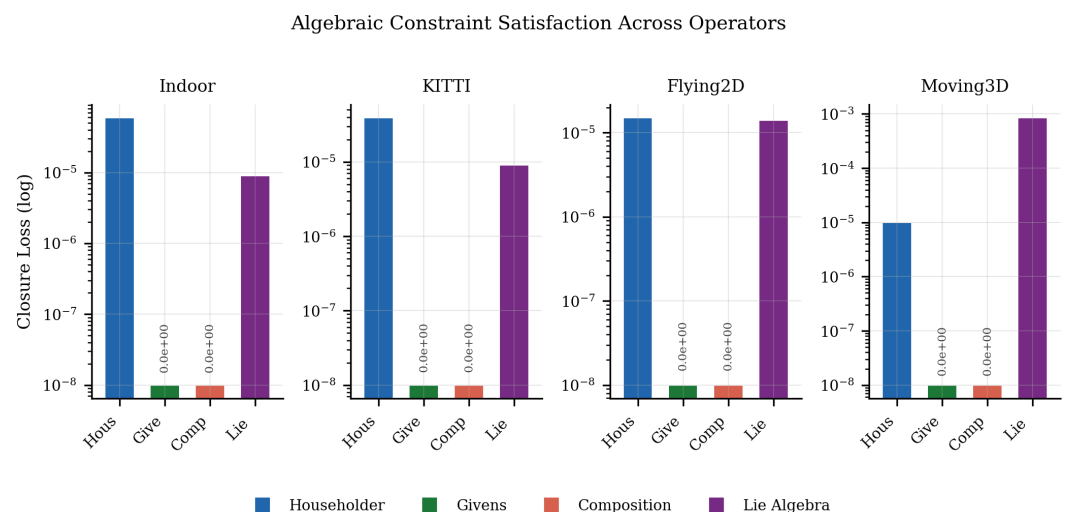


Figure A7. Algebraic closure loss at $\sigma = 0$ for all operators and datasets.

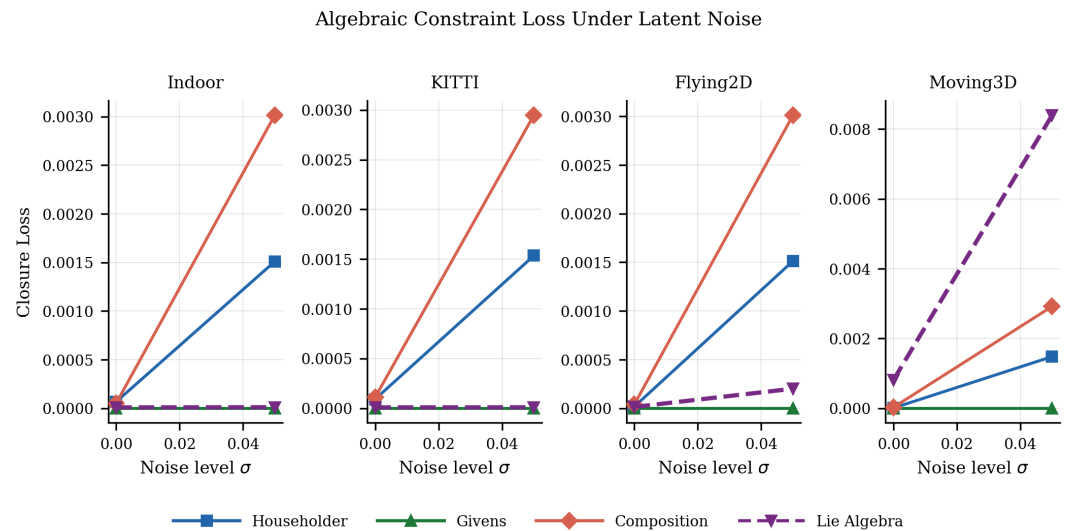


Figure A8. Algebraic closure loss under latent Gaussian noise across all datasets.

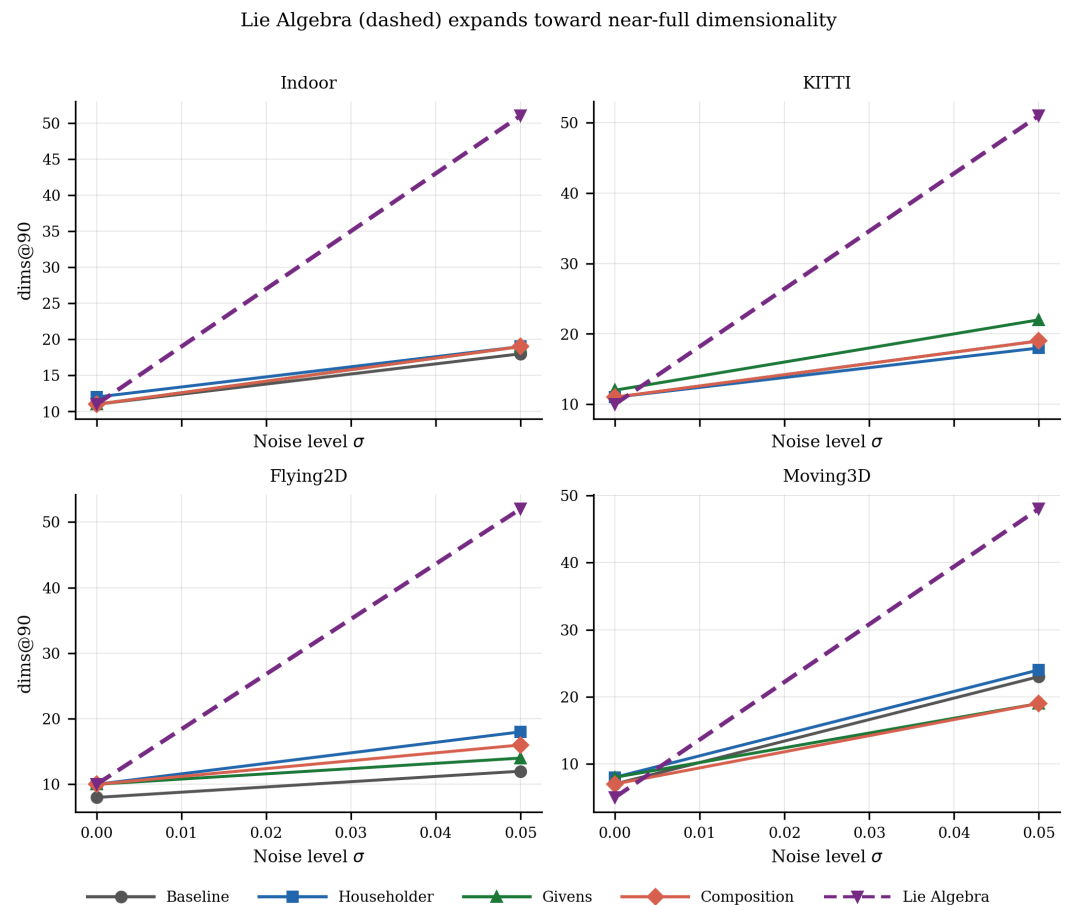


Figure A9. Dims@90 as a function of latent noise level σ .

We report bootstrap confidence intervals only for entropy and activation variance. These two metrics are computed from per-dimension activation statistics and resample cleanly from the held-out hidden-state arrays. Dims@90 and effective rank are matrix-level descriptors of the full population matrix H_7 and are reported from the primary evaluation pass in Table 7; their perturbation sensitivity is assessed through the noise sweep in Table 8.

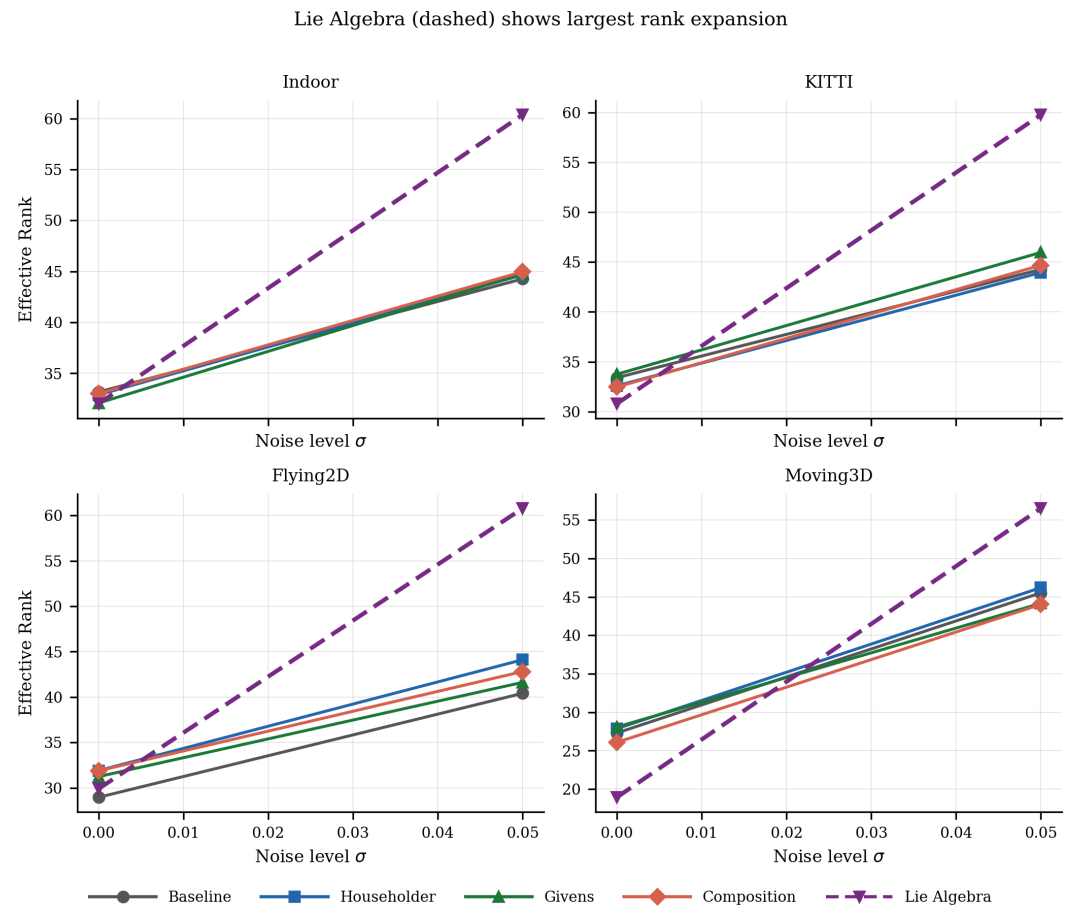


Figure A10. Effective rank as a function of latent noise level σ .

Table A1. Bootstrap 95% confidence intervals for entropy and activation variance ($n_{\text{boot}} = 2000$ resamples of the held-out hidden-state arrays used for Table 7). Each cell reports the observed value [95 % CI lower, upper]. **Bold** indicates that the operator CI does not overlap with the baseline CI for the same dataset and metric.

Dataset	Operator	Entropy	Act. Var
Indoor	Baseline	5.2607 [5.2591, 5.2622]	0.0329 [0.0327, 0.0332]
	ConvLSTM-cell	5.5934 [5.5897, 5.5969]	0.0305 [0.0302, 0.0307]
	Householder	5.3538 [5.3516, 5.3560]	0.0311 [0.0309, 0.0313]
	Givens	5.2523 [5.2486, 5.2561]	0.0286 [0.0284, 0.0288]
	Composition	5.3806 [5.3786, 5.3824]	0.0309 [0.0307, 0.0311]
	Lie Algebra	5.3692 [5.3656, 5.3725]	0.0024 [0.0024, 0.0024]
KITTI	Baseline	5.3412 [5.3316, 5.3499]	0.0254 [0.0246, 0.0261]
	ConvLSTM-cell	5.4998 [5.4897, 5.5087]	0.0545 [0.0535, 0.0555]
	Householder	5.4231 [5.4172, 5.4286]	0.0261 [0.0255, 0.0268]
	Givens	5.2344 [5.2268, 5.2412]	0.0202 [0.0197, 0.0207]
	Composition	5.2109 [5.2029, 5.2188]	0.0233 [0.0228, 0.0239]
	Lie Algebra	5.1094 [5.0905, 5.1264]	0.0016 [0.0016, 0.0017]
Flying2D	Baseline	5.6579 [5.6500, 5.6646]	0.0345 [0.0338, 0.0352]
	ConvLSTM-cell	5.6942 [5.6881, 5.6990]	0.0271 [0.0267, 0.0274]
	Householder	5.5571 [5.5491, 5.5649]	0.0250 [0.0243, 0.0256]
	Givens	5.7002 [5.6906, 5.7084]	0.0347 [0.0339, 0.0353]
	Composition	5.6312 [5.6217, 5.6389]	0.0303 [0.0296, 0.0309]
	Lie Algebra	5.2831 [5.2722, 5.2929]	0.0014 [0.0013, 0.0014]
Moving3D	Baseline	5.4309 [5.4287, 5.4329]	0.0150 [0.0149, 0.0151]
	ConvLSTM-cell	5.2930 [5.2896, 5.2970]	0.0122 [0.0121, 0.0123]
	Householder	5.2912 [5.2882, 5.2938]	0.0149 [0.0148, 0.0151]
	Givens	5.5037 [5.5011, 5.5060]	0.0190 [0.0188, 0.0191]
	Composition	5.3562 [5.3538, 5.3585]	0.0178 [0.0177, 0.0179]
	Lie Algebra	4.9215 [4.9192, 4.9237]	0.0028 [0.0027, 0.0028]

Appendix A.4. Image-Space Perturbation Robustness

To complement the latent-noise analysis, we also evaluate supplementary pixel-space perturbations on the Indoor held-out set, including brightness, contrast, and blur. Blur produces only minor MAE changes for all variants, including the baseline, suggesting that blur tolerance is mainly associated with the shared encoder–decoder architecture. Brightness and contrast shifts are more disruptive because they alter the input distribution before the encoder. We therefore report this perturbation test on the Indoor held-out set using brightness, contrast, and blur as a supplementary diagnostic because this paper’s main claim concerns latent-space reorganization under a strict bottleneck, not general image-corruption robustness.

Table A2. Image-space perturbation robustness on the Indoor held-out test set. Perturbations are applied to the input sequence only, while the ground-truth target remains unperturbed. Brightness, contrast, and blur are reported at the strongest tested severity. $\uparrow$ higher is better; $\downarrow$ lower is better. $\Delta\text{MAE} = \text{MAE}(\text{perturbed}) - \text{MAE}(\text{clean})$.

Model	Perturbation	PSNR $\uparrow$	MAE $\downarrow$	ΔMAE	Entropy	Dims@90	Eff. Rank	Act. Var
Baseline	Clean	17.412	0.1045	0.0000	5.2607	18	42.933	0.0329
	Brightness	11.261	0.2331	+0.1286	4.9649	19	42.296	0.0116
	Contrast	15.796	0.1347	+0.0302	5.2246	13	37.957	0.0118
	Blur	17.369	0.1061	+0.0016	5.2881	16	40.883	0.0313
Householder	Clean	18.084	0.0952	0.0000	5.3538	20	43.467	0.0311
	Brightness	10.600	0.2508	+0.1556	5.0763	15	36.748	0.0160
	Contrast	16.765	0.1179	+0.0227	5.2168	16	39.002	0.0118
	Blur	18.154	0.0954	+0.0002	5.3577	19	42.547	0.0296
Givens	Clean	18.180	0.0937	0.0000	5.2523	19	42.595	0.0286
	Brightness	10.949	0.2450	+0.1512	5.2350	16	37.674	0.0158
	Contrast	16.767	0.1188	+0.0251	5.0476	15	37.743	0.0098
	Blur	18.255	0.0938	+0.0001	5.2475	18	41.771	0.0275
Composition	Clean	18.094	0.0948	0.0000	5.3806	20	43.335	0.0309
	Brightness	11.278	0.2348	+0.1401	5.1705	16	38.353	0.0143
	Contrast	16.734	0.1196	+0.0248	5.2382	16	40.004	0.0113
	Blur	18.192	0.0946	−0.0002	5.3736	19	42.495	0.0288
Lie Algebra	Clean	18.016	0.0965	0.0000	5.3692	17	41.787	0.0024
	Brightness	10.605	0.2537	+0.1572	4.2792	14	35.609	0.0020
	Contrast	16.340	0.1266	+0.0302	5.3972	14	37.514	0.0005
	Blur	18.097	0.0969	+0.0005	5.3489	17	41.072	0.0020

Appendix A.5. Fixed-Point Weight Sensitivity

To check whether the Indoor results are tied to the default fixed-point weight, we evaluate $\lambda_{\text{fp}} \in \{0.5, 1.0, 2.0\}$ for Householder, Givens, and Lie Algebra. The sweep tests sensitivity to regularization strength rather than optimizing λ_{fp} .

The sensitivity results support two observations. First, reconstruction quality is relatively stable across the tested range: the maximum PSNR variation is 0.25 dB, observed for Lie Algebra, and $\lambda_{\text{fp}} = 1.0$ is not the worst-performing setting for any operator. Second, the lightweight operators preserve stable latent participation, with Dims@90 unchanged for Givens and changing by only one dimension for Householder. Lie Algebra shows a different pattern: activation variance decreases monotonically as λ_{fp} increases, but it is already strongly reduced at $\lambda_{\text{fp}} = 0.5$. This indicates that the over-compression observed for Lie Algebra is not simply an artifact of the default $\lambda_{\text{fp}} = 1.0$ setting but reflects the stronger constraint imposed by the full $so(k)$ parameterization in this bottleneck architecture.

Table A3. Indoor reconstruction sensitivity to the fixed-point weight λ_{fp} on the full held-out set. $\uparrow$ higher is better; $\downarrow$ lower is better.

Operator	λ_{fp}	PSNR $\uparrow$	SSIM $\uparrow$	MAE $\downarrow$
Householder	0.5	17.9462	0.5387	0.09714
Householder	1.0	18.0841	0.5414	0.09521
Householder	2.0	17.9545	0.5368	0.09692
Givens	0.5	18.2328	0.5459	0.09332
Givens	1.0	18.1796	0.5449	0.09374
Givens	2.0	18.0111	0.5437	0.09579
Lie Algebra	0.5	18.0944	0.5399	0.09539
Lie Algebra	1.0	18.0160	0.5409	0.09647
Lie Algebra	2.0	17.8408	0.5381	0.09775

Table A4. Indoor latent-metric sensitivity to the fixed-point weight λ_{fp} on the full held-out set.

Operator	λ_{fp}	Entropy	Dims@90	Eff. Rank	Act. Var
Householder	0.5	5.4503	20	43.426	0.040268
Householder	1.0	5.3538	20	43.467	0.031143
Householder	2.0	5.4181	21	44.005	0.032269
Givens	0.5	5.4150	19	43.009	0.034764
Givens	1.0	5.2523	19	42.595	0.028625
Givens	2.0	5.2351	19	42.804	0.026930
Lie Algebra	0.5	5.1927	18	42.003	0.004058
Lie Algebra	1.0	5.3692	17	41.787	0.002407
Lie Algebra	2.0	5.5200	19	42.466	0.000435

References

1. Su, W.; Zhu, X.; Tao, C.; Lu, L.; Li, B.; Huang, G.; Qiao, Y.; Wang, X.; Zhou, J.; Dai, J. Towards All-in-One Pre-Training via Maximizing Multi-Modal Mutual Information. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 15888–15899. [\[CrossRef\]](#)
2. Shinya, Y. USB: Universal-Scale Object Detection Benchmark. In Proceedings of the 33rd British Machine Vision Conference (BMVC), London, UK, 21–24 November 2022.
3. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:2010.11929.
4. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models Are Few-Shot Learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
5. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. PaLM: Scaling Language Modeling with Pathways. *J. Mach. Learn. Res.* **2023**, *24*, 113.
6. Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; Van Den Driessche, G.B.; Lespiau, J.B.; Damoc, B.; Clark, A.; et al. Improving Language Models by Retrieving from Trillions of Tokens. In Proceedings of the 39th International Conference on Machine Learning (ICML), Baltimore, MD, USA, 17–23 July 2022; pp. 2206–2240.
7. Schierholz, M.; Yang, C.; Roy, K.; Swaminathan, M.; Schuster, C. Comparison of Collaborative Versus Extended Artificial Neural Networks for PDN Design. In Proceedings of the 2020 IEEE 24th Workshop on Signal and Power Integrity (SPI), Cologne, Germany, 17–20 May 2020; pp. 1–4. [\[CrossRef\]](#)
8. Roy, K.; Mert, H.T.; Swaminathan, M. Preliminary Application of Deep Learning to Design Space Exploration. In Proceedings of the 2018 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS), Chandigarh, India, 16–18 December 2018; pp. 1–3. [\[CrossRef\]](#)
9. Patel, R.; Roy, K.; Choi, J.; Han, K.J. Generative Design of Electromagnetic Structures Through Bayesian Learning. *IEEE Trans. Magn.* **2018**, *54*, 1–4. [\[CrossRef\]](#)
10. Ostendorff, M.; Rethmeier, N.; Augenstein, I.; Gipp, B.; Rehm, G. Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP), Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 11670–11688. [\[CrossRef\]](#)
11. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.

12. Cayton, L. *Algorithms for Manifold Learning*; Technical Report; Research Examination Paper; University of California: San Diego, CA, USA, 2005.
13. Brahma, P.P.; Wu, D.; She, Y. Why Deep Learning Works: A Manifold Disentanglement Perspective. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 1997–2008. [[CrossRef](#)] [[PubMed](#)]
14. Monti, F.; Boscaini, D.; Masci, J.; Rodolà, E.; Svoboda, J.; Bronstein, M.M. Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5115–5124.
15. Bronstein, M.M.; Bruna, J.; Cohen, T.; Veličković, P. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *arXiv* **2021**, arXiv:2104.13478.
16. Monti, F.; Frasca, F.; Eynard, D.; Mannion, D.; Bronstein, M.M. Fake News Detection on Social Media Using Geometric Deep Learning. *arXiv* **2019**, arXiv:1902.06673.
17. Winter, R.; Noé, F.; Clevert, D.A. Permutation-Invariant Variational Autoencoder for Graph-Level Representation Learning. In Proceedings of the Advances in Neural Information Processing Systems, Virtual Event, 6–14 December 2021.
18. Duan, T.; Lee, J. Graph Embedding VAE: A Permutation Invariant Model of Graph Structure. *arXiv* **2019**, arXiv:1910.08057.
19. Hansen, A.A.; Calissano, A.; Feragen, A. Interpreting Equivariant Representations. In Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024.
20. Thiede, E.H.; Hy, T.S.; Kondor, R. The General Theory of Permutation Equivariant Neural Networks and Higher Order Graph Variational Encoders. *arXiv* **2020**, arXiv:2004.03990.
21. Kortvelesy, R.; Morad, S.; Prorok, A. Permutation-Invariant Set Autoencoders with Fixed-Size Embeddings for Multi-Agent Learning. In *AAMAS '23: Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*; International Foundation for Autonomous Agents and Multiagent Systems: Richland, SC, USA, 2023.
22. Stirling, J.S.J.; Al-Moubayed, N.; Shum, H.P.H. Investigating Permutation-Invariant Discrete Representation Learning for Spatially Aligned Images. *arXiv* **2026**, arXiv:2604.01843.
23. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015.
24. Wang, Y.; Long, M.; Wang, J.; Gao, Z.; Yu, P.S. PredRNN: Recurrent Neural Networks for Predictive Learning Using Spatiotemporal LSTMs. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
25. Wang, Y.; Gao, Z.; Long, M.; Wang, J.; Yu, P.S. PredRNN++: Towards A Resolution of the Deep-in-Time Dilemma in Spatiotemporal Predictive Learning. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5123–5132.
26. Wang, Y.; Jiang, L.; Yang, M.H.; Li, L.J.; Long, M.; Fei-Fei, L. Eidetic 3D LSTM: A Model for Video Prediction and Beyond. In Proceedings of the 7th International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
27. Arjovsky, M.; Shah, A.; Bengio, Y. Unitary Evolution Recurrent Neural Networks. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1120–1128.
28. Mhammedi, Z.; Hellicar, A.; Rahman, A.; Bailey, J. Efficient Orthogonal Parametrisation of Recurrent Neural Networks Using Householder Reflections. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2401–2409.
29. Cohen, T.; Welling, M. Group Equivariant Convolutional Networks. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 2990–2999.
30. Weiler, M.; Cesa, G. General E(2)-Equivariant Steerable CNNs. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
31. Simard, P.Y.; LeCun, Y.A.; Denker, J.S.; Victorri, B. Transformation Invariance in Pattern Recognition: Tangent Distance and Tangent Propagation. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 30 November–3 December 1992; pp. 857–864.
32. Rifai, S.; Vincent, P.; Muller, X.; Glorot, X.; Bengio, Y. Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. In Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA, 28 June–2 July 2011; pp. 833–840.
33. Sajjadi, M.; Javanmardi, M.; Tasdizen, T. Regularization with Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
34. Laine, S.; Aila, T. Temporal Ensembling for Semi-Supervised Learning. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
35. Tarvainen, A.; Valpola, H. Mean Teachers Are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-Supervised Deep Learning Results. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 1195–1204.

36. Miyato, T.; Maeda, S.i.; Koyama, M.; Ishii, S. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1979–1993. [[CrossRef](#)] [[PubMed](#)]
37. Ke, Y.; Li, X.; Liang, Y.; Shi, Z.; Song, Z. Advancing the Understanding of Fixed Point Iterations in Deep Neural Networks: A Detailed Analytical Study. *arXiv* **2024**, arXiv:2410.11279.
38. Movahedi, S.; Feigin, S.L.; Milovanović, V.; Theus, A.; Hofmann, T.; Boeva, V.; Rusch, T.K.; Orvieto, A. Fixed-Point Reasoning: Stable and Adaptive Deep Looped Models. In Proceedings of the Workshop on High-dimensional Learning Dynamics (HiLD), ICML 2026, Seoul, Republic of Korea, 10 July 2026.
39. Lin, D.D.; Talathi, S.S.; Annapureddy, V.S. Fixed Point Quantization of Deep Convolutional Networks. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 2849–2858.
40. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
41. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision Meets Robotics: The KITTI Dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
42. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
43. Huang, Y.W.; Wang, W. PVMNet: A navel orange defect detection algorithm based on Mamba structure. *Expert Syst. Appl.* **2026**, *300*, 130412. [[CrossRef](#)]
44. Yu, X.; Liang, X.; Zhou, Z.; Zhang, B. Multi-task learning for hand heat trace time estimation and identity recognition. *Expert Syst. Appl.* **2024**, *255*, 124551. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.