

ST-GRU PredRNN-V2: A GRU-Based Reformulation of PredRNN-V2 for Video Prediction

Augusta van Haren
Radboud University

Hanna Hoogen
Radboud University

Luca Pattavina
Radboud University

Quincy Rommen
Radboud University

October 2026

All authors contributed equally.

Abstract

Video prediction requires models to capture complex spatiotemporal dependencies and to retain relevant information over long sequences. PredRNN and its successor PredRNN-V2 address this challenge with spatiotemporal Long Short-Term Memory (ST-LSTM) units that maintain two separate memory states, at the cost of a relatively complex recurrent building block.

In this work, we investigate whether the simpler gating of Gated Recurrent Units (GRUs) can be transferred to spatiotemporal predictive learning. To this end, we develop a spatiotemporal GRU (ST-GRU) unit as a GRU-based alternative to the ST-LSTM. The ST-GRU replaces the ST-LSTM’s three gates per memory state with two GRU-style gates (update and reset), and merges the temporal memory state into the hidden state, which reduces the number of parameters by about 5%. We incorporate the ST-GRU into PredRNN-V2 and evaluate the resulting model on the Moving MNIST video prediction benchmark, following the original experimental protocol.

Averaged over ten predicted frames, the ST-GRU-based PredRNN-V2 achieves an MSE of 55.1, an SSIM of 0.883, an LPIPS of 0.065 and a PSNR of 19.53dB. It does not surpass the original ST-LSTM-based PredRNN-V2 on MSE, SSIM or PSNR, but it achieves a lower LPIPS and performs on par with the earlier ST-LSTM-based PredRNN, while using a simpler recurrent cell. These results suggest that ST-GRUs are a promising building block for spatiotemporal predictive learning and motivate further work on their design and integration.

1 Introduction

Recent developments in Artificial Intelligence (AI) have enabled the synthetic creation of images with remarkable success (Betker et al., 2023). Generating video frames, however, remains challenging, due to complex spatiotemporal dependencies within the data and uncertainty in predicting future motion (Cai et al., 2018). This task, known as predictive learning, focuses on generating future data points conditioned on data points from the past. For video, the aim is to predict upcoming frames based on a sequence of past frames.

Modelling sequential data was a key motivation for developing recurrent neural networks (RNNs), which capture temporal dependencies by maintaining a hidden state over time. However, RNNs suffer from a fundamental bottleneck: the vanishing gradient problem limits their ability to retain long-term dependencies, particularly in long input sequences (Hochreiter, 1998). As a consequence, the network forgets information that is relevant for its future predictions, which reduces their accuracy.

Two gated architectures were developed to alleviate this problem: Long Short-Term Memory (LSTM) units and Gated Recurrent Units (GRUs). The LSTM adds a memory cell controlled by three gates (an input gate, a forget gate and an output gate), which together decide what information to store, discard and expose. This allows the network to retain temporal dependencies over much longer time spans (Hochreiter and Schmidhuber, 1997). The GRU is a later development that follows the same principle of selective information retention, but simplifies the architecture: it has only an update gate and a reset gate, and no separate memory cell (Chung et al., 2014).

LSTMs and GRUs have been widely used for modelling sequential data such as language and speech. Video prediction, however, additionally requires models to preserve spatial structure, representing both low-level appearance and high-level dynamics in order to generate future frames. Convolutional variants such as ConvLSTM (Shi et al., 2015) and ConvGRU (Ballas et al., 2016) address this by replacing the fully connected transformations inside the recurrent units with convolutions.

Building on this, Wang et al. (2017) developed the Predictive Recurrent Neural Network (PredRNN), which was later extended to PredRNN-V2 (Wang et al., 2021). Instead of classical LSTMs, PredRNN is built on

spatiotemporal LSTM (ST-LSTM) units that maintain not one but *two* memory states. The way these ST-LSTM units are chained allows spatiotemporal information to flow both across time steps and across the layers of the network. Through this mechanism, top-down expectations and bottom-up signals are combined to predict future inputs. This differs from stacked ConvLSTMs, in which memory states are confined to individual recurrent units. PredRNN(-V2) outperformed previous architectures on several video prediction benchmarks.

Although PredRNN(-V2) predicts spatiotemporal dynamics well across various datasets, its ST-LSTM units are relatively complex, with three gates per memory state and two separate memory states. GRUs, in contrast, use fewer gates than LSTMs while typically achieving comparable performance on sequential tasks (Chung et al., 2014). This work explores whether this simplification also transfers to spatiotemporal predictive learning.

To this end, we develop a GRU-based alternative to the ST-LSTM, which we call the spatiotemporal GRU (ST-GRU). We replace PredRNN-V2’s ST-LSTM units with ST-GRUs and compare the resulting model with the original PredRNN-V2, as well as with several other recurrent baselines, on video frame prediction on the Moving MNIST dataset. Concurrently with our work, Dong et al. (2025) proposed a different ST-GRU cell for the original PredRNN; we include their results in our comparison and discuss how the two approaches differ. Our code is available at <https://github.com/xPatta/ST-GRU>.

2 Methods

2.1 ST-LSTM-based PredRNN

PredRNN (Wang et al., 2017) consists of a stack of ST-LSTMs with a dual-memory mechanism that learns both the spatial structure and the temporal dynamics of sequential data, allowing it to predict future frames.

The ST-LSTMs in PredRNN handle four flows of information: the input state $\mathcal{X}_t$, the hidden state $\mathcal{H}_t^l$, and *two* memory states, $\mathcal{C}_t^l$ and $\mathcal{M}_t^l$. These memory states are decoupled. The temporal memory $\mathcal{C}_t^l$ operates on a slower timescale and is designed to learn long-term dependencies in the data. It is passed horizontally from time step $t - 1$ to time step t within the same layer l . The spatiotemporal memory $\mathcal{M}_t^l$, in contrast, adapts more rapidly to changes in the input and captures short-term dynamics. It follows a deeper transition path that zigzags upwards, passing information from layer $l - 1$ to layer l within the same time step; only the top layer at time step $t - 1$ is connected to the bottom layer at time step t .

Each memory state has its own input gate i_t (i_t^l), forget gate f_t (f_t^l) and input modulation gate g_t (g_t^l). The input gate, with a sigmoid activation (σ), determines how much new information from the current input is added to the memory. The forget gate decides what information is discarded. The input modulation gate generates candidate values using the hyperbolic tangent ($\tanh$); these are scaled by the input gate as $i_t \odot g_t$ and added to the memory state, replacing the forgotten information.

Finally, the updated memory states $\mathcal{C}_t^l$ and $\mathcal{M}_t^l$ produce the new hidden state $\mathcal{H}_t^l$ in two ways: indirectly via the output gate o_t , and directly via concatenation, a 1×1 convolution and a $\tanh$. The ST-LSTM equations are given in Figure 4a, and schematics of the cell and its integration in PredRNN(-V2) are shown in Figures 5a and 6a (Appendix A).

2.2 ST-LSTM-based PredRNN-V2

PredRNN-V2 (Wang et al., 2021) extends PredRNN in three ways. First, a decoupling loss explicitly increases the distance between the updates of the two memory states in latent space, encouraging them to learn different aspects of the spatiotemporal variations and preventing redundant features. Second, reverse scheduled sampling is a curriculum learning strategy for the encoder that gradually shifts its input from the model’s own predictions to the ground-truth frames, which forces the model to learn long-term dynamics. Third, PredRNN-V2 supports action-conditioned video prediction.

2.3 ST-GRU-based PredRNN-V2

We propose a variant of PredRNN-V2 in which the ST-LSTM building blocks are replaced by GRU-based counterparts, which we call ST-GRUs. We design the ST-GRU around the key characteristics of a GRU, while keeping the separation between temporal and spatiotemporal memory flows of the ST-LSTM (Wang et al., 2017, 2021).

Following the GRU, each memory update in the ST-GRU uses only two gates: an update gate z_t (respectively z_t^l) and a reset gate r_t (respectively r_t^l). In addition, the ST-GRU omits the separate temporal memory state $\mathcal{C}_t^l$ and integrates its function into the hidden state $\mathcal{H}_t^l$, since both follow the same horizontal path from one time step to the next.

As a result, the two pairs of gates act on $\mathcal{H}_t^l$ and $\mathcal{M}_t^l$, rather than on $\mathcal{C}_t^l$ and $\mathcal{M}_t^l$ as in the ST-LSTM. The update gate z_t decides how much of the state is replaced by new content, whereas the reset gate r_t decides how much of the previous state is used when computing that new content. Together they play roles similar to the

input and forget gates of the ST-LSTM (Chung et al., 2014). The gates of both states share the convolutions applied to $\mathcal{M}_t^{l-1}$, which further reduces the number of parameters. The gates are used to compute the candidate states $\tilde{\mathcal{H}}_t^l$ and $\tilde{\mathcal{M}}_t^l$, which then produce the new hidden state $\mathcal{H}_t^l$ indirectly via the output gate o_t , and directly via concatenation, a 1×1 convolution and a tanh. The ST-GRU equations are given in Figure 4b, and schematics of the cell and its integration in PredRNN(-V2) are shown in Figures 5b and 6b (Appendix A), next to their ST-LSTM counterparts. The figures highlight both what the two cells share (the decoupled spatiotemporal memory) and where they differ (the number of gates and memory states).

2.4 Implementation

We implemented the ST-GRU-based PredRNN-V2 on top of the official PyTorch implementation of PredRNN-V2 (Wu, 2022). Following the structure of the original ST-LSTM cell, we implemented the ST-GRU as designed in Section 2.3, with two instead of three gates per memory state and without using $\mathcal{C}_t^l$. The variable for $\mathcal{C}_t^l$ is retained in the code only to keep the interface compatible with PredRNN-V2. We then built a version of PredRNN-V2 that uses the ST-GRU as its building block, passing information through ST-GRUs instead of ST-LSTMs. Because the ST-GRU has no temporal memory update to decouple from $\mathcal{M}_t^l$, we trained this model without PredRNN-V2’s decoupling loss, using only the MSE loss; reverse scheduled sampling was kept unchanged. All other settings follow the official Moving MNIST configuration: four recurrent layers with 128 hidden channels each, 5×5 convolutions, a patch size of 4, a batch size of 8 and a learning rate of 10^{-4} . With these settings, the ST-GRU-based PredRNN-V2 has 22.6M parameters, compared with 23.9M for the ST-LSTM-based PredRNN-V2, a reduction of about 5%. Our implementation is available at <https://github.com/xPatta/ST-GRU>.

2.5 Experimental Setup

To compare the ST-GRU-based PredRNN-V2 directly with the original ST-LSTM-based version, we repeated the Moving MNIST experiment of Wang et al. (2021), following the same training and testing protocol. Specifically, we trained the model for 80,000 iterations using the official training configuration (Wu, 2022), and evaluated it with four metrics: mean squared error (MSE), structural similarity index measure (SSIM) (Wang et al., 2004), learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018), and peak signal-to-noise ratio (PSNR).

We further compared our model with several other recurrent models. ConvLSTM (Shi et al., 2015) and ConvGRU (Ballas et al., 2016) are convolutional LSTM- and GRU-based RNNs *without* decoupled spatiotemporal memory. The ST-LSTM-based PredRNN (Wang et al., 2017) has decoupled spatiotemporal memory but *without* the V2 additions (see Section 2.2). Finally, Dong et al. (2025) concurrently proposed an ST-GRU cell and integrated it into PredRNN. Because they evaluated it with the same experimental protocol as above, we include their reported results in our comparison. We emphasise that their ST-GRU-based PredRNN differs from our ST-GRU-based PredRNN-V2 *both* in the design of the ST-GRU cell and in the absence of the V2 additions (see Section 2.2). Results for the LSTM-based baselines are taken from Wang et al. (2021), and those for the GRU-based baselines from Dong et al. (2025). PSNR is not reported in these sources, so we computed it for the ST-LSTM-based PredRNN-V2 using the official trained model (Wu, 2022).

3 Results

As shown in Figure 1, all four frame-wise metrics (MSE, SSIM, LPIPS and PSNR) plateau within the 80,000 training iterations, indicating that the number of iterations was sufficient for convergence. Early frames are consistently predicted more accurately than later frames. This is expected, as early frames are temporally closer to the input sequence and are therefore inherently easier to predict.

Visual inspection of the generated frames confirms that the model predicts earlier frames (i.e., frames closer to the input frames) more accurately (Figure 3). Local image features degrade gradually in later frames, especially when the trajectories of the two digits overlap.

3.1 Quantitative Model Comparison

Table 1 summarises the performance of all models in terms of MSE, SSIM, LPIPS and PSNR, averaged over the ten predicted frames. On MSE and SSIM, the ST-GRU-based PredRNN-V2 performs on par with the ST-LSTM-based PredRNN (MSE 55.1 vs. 56.8; SSIM 0.883 vs. 0.867). It clearly outperforms ConvLSTM on all metrics, and ConvGRU on SSIM and LPIPS, while its MSE is only slightly lower than that of ConvGRU. The ST-LSTM-based PredRNN-V2 and the ST-GRU-based PredRNN of Dong et al. (2025) achieve better MSE and SSIM than our model. On LPIPS, the ST-GRU-based PredRNN-V2 *does* outperform the ST-LSTM-based

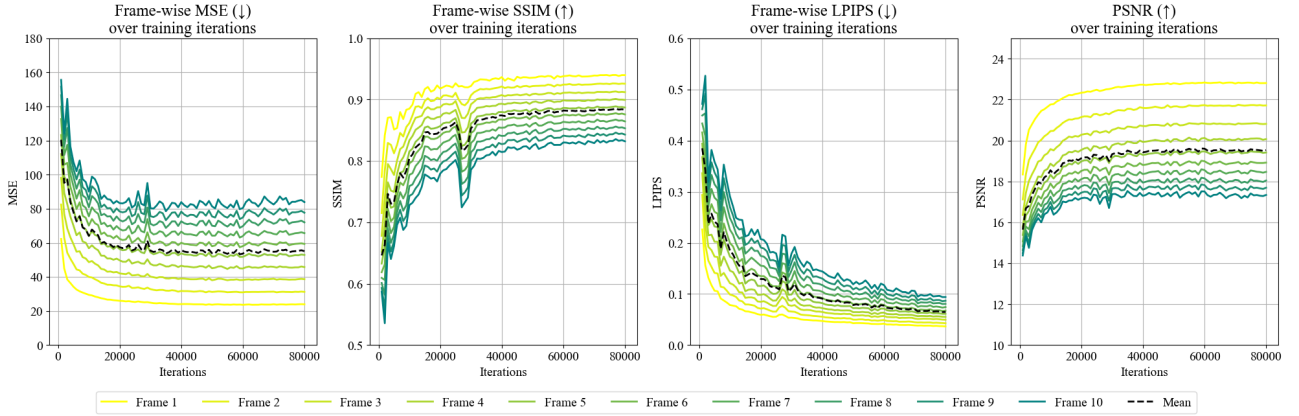


Figure 1: Frame-wise metrics (MSE, SSIM, LPIPS, PSNR) of the ST-GRU-based PredRNN-V2 on the Moving MNIST test set over 80,000 training iterations. The frame number indicates the number of time steps between the predicted frame and the last observed frame.

PredRNN-V2 (0.065 vs. 0.071), and is surpassed only by the ST-GRU-based PredRNN (0.057). On PSNR, which we computed only for the two PredRNN-V2 variants, the ST-LSTM-based model scores higher (20.05 vs. 19.53 dB).

Model	MSE ($\downarrow$)	SSIM ($\uparrow$)	LPIPS ($\downarrow$)	PSNR ($\uparrow$)
ConvLSTM (RNN) (Shi et al., 2015)	103.3	0.707	0.156	–
ST-LSTM (PredRNN) (Wang et al., 2017)	56.8	0.867	0.107	–
ST-LSTM (PredRNN-V2) (Wang et al., 2021)	48.4	0.891	0.071	20.05
ConvGRU (RNN) (Ballas et al., 2016)	57.8	0.839	0.091	–
ST-GRU (PredRNN) (Dong et al., 2025)	53.4	0.893	0.057	–
ST-GRU (PredRNN-V2), ours	55.1	0.883	0.065	19.53

Table 1: Model performance on the Moving MNIST test set. All metrics are averaged over ten predicted frames. The best result for each metric is shown in bold, and our model in teal. Results for the LSTM-based baselines are taken from Wang et al. (2021) and for the GRU-based baselines from Dong et al. (2025). PSNR (in dB) was computed only for the two PredRNN-V2 variants.

The frame-wise results in Figure 2 agree with these averages. In MSE and SSIM, the ST-GRU-based PredRNN-V2 closely tracks the ST-LSTM-based PredRNN across all ten frames. Frame-wise LPIPS is only available for the LSTM-based baselines; compared with these, our model achieves the lowest LPIPS, and its advantage over the ST-LSTM-based PredRNN-V2 grows for later frames. In PSNR, the ST-LSTM-based PredRNN-V2 is consistently higher, in line with Table 1.

4 Discussion

Replacing the ST-LSTM with our ST-GRU in PredRNN-V2 yields a model that performs well across metrics and frames, but that is generally outperformed by the original ST-LSTM-based PredRNN-V2. We believe this is largely due to how the improvements that define PredRNN-V2 interact with our design. The decoupling loss acts on the two separate memory states $\mathcal{C}_t^l$ and $\mathcal{M}_t^l$. Because the ST-GRU does not use $\mathcal{C}_t^l$, this loss cannot be applied in its original form, and we trained the ST-GRU-based model without it. As our experiments did not involve action-conditioned video prediction, the ST-GRU-based PredRNN-V2 therefore benefits mainly from reverse scheduled sampling, whereas the ST-LSTM-based PredRNN-V2 benefits from both reverse scheduled sampling and the decoupling loss. Our results suggest that the simplified cell does not compensate for the loss of this mechanism. Adapting the decoupling loss to act between $\mathcal{H}_t^l$ and $\mathcal{M}_t^l$ is therefore a natural next step.

Interestingly, our model achieves a lower LPIPS than the ST-LSTM-based PredRNN-V2, especially for later frames. LPIPS measures distance in the feature space of a pretrained deep network and is designed to reflect human judgements of perceptual similarity (Zhang et al., 2018). This suggests that the ST-GRU-based model preserves perceptually relevant structure better, even when its pixel-wise error is higher.

The concurrent ST-GRU-based PredRNN of Dong et al. (2025) outperforms our model on all reported metrics, even though it does not use the PredRNN-V2 improvements. This indicates that the specific design of the

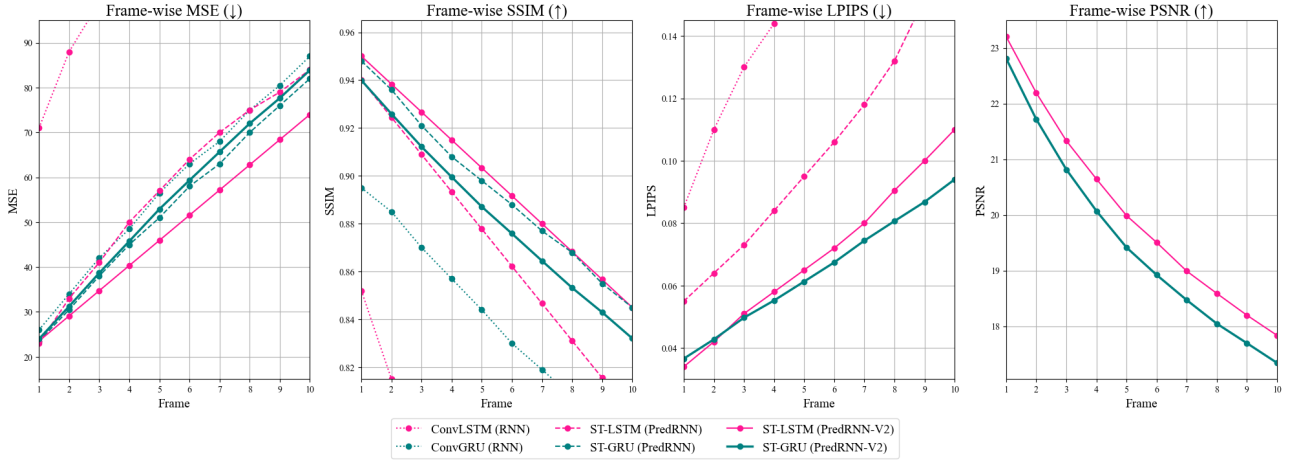


Figure 2: Frame-wise MSE, SSIM, LPIPS and PSNR on the Moving MNIST test set. All models were trained and evaluated following the protocol described in Section 2.5. Results for the LSTM-based models are taken from Wang et al. (2021), and those for the GRU-based models from Dong et al. (2025). PSNR for the ST-LSTM-based PredRNN-V2 was computed with the official trained model (Wu, 2022). All results for the ST-GRU-based PredRNN-V2 are from our own evaluation.

gated cell matters at least as much as these training improvements. Nevertheless, our ST-GRU-based PredRNN-V2 performs on par with the ST-LSTM-based PredRNN while using a simpler building block. Comparing the two ST-GRU designs under identical conditions, and combining the stronger cell with the PredRNN-V2 training scheme, are promising directions for future work.

Limitations. We evaluated our model only on Moving MNIST, a synthetic benchmark; experiments on more complex, real-world datasets are needed to establish how well the ST-GRU generalises. We report results from a single training run. Results for the baselines are taken from the literature rather than re-trained under identical conditions. Finally, the ST-GRU reduces the parameter count by only about 5%, and we did not measure training time, inference time or memory use, so its practical computational benefits remain to be quantified.

	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Frame 8	Frame 9	Frame 10
Input sequence										
True future sequence										
Predicted future sequence										

	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Frame 8	Frame 9	Frame 10
Input sequence										
True future sequence										
Predicted future sequence										

	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Frame 8	Frame 9	Frame 10
Input sequence										
True future sequence										
Predicted future sequence										

	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Frame 8	Frame 9	Frame 10
Input sequence										
True future sequence										
Predicted future sequence										

	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Frame 8	Frame 9	Frame 10
Input sequence										
True future sequence										
Predicted future sequence										

Figure 3: Predictions of the ST-GRU-based PredRNN-V2 on five Moving MNIST test sequences, showing the input sequence, the ground-truth future sequence and the predicted future sequence. Both global and local similarity to the ground truth decline for frames further into the future.

5 Conclusion

We presented the ST-GRU-based PredRNN-V2, a reformulation of PredRNN-V2 (Wang et al., 2021) that replaces its ST-LSTM building blocks with a newly designed ST-GRU cell. The ST-GRU keeps the decoupled spatiotemporal memory flow of the ST-LSTM, but uses fewer gates, one fewer memory state and about 5% fewer parameters. On Moving MNIST, the model successfully captures the spatiotemporal dynamics of ten-frame future sequences, with accuracy decreasing for later frames, as expected. It performs on par with the ST-LSTM-based PredRNN (Wang et al., 2017) and achieves a lower LPIPS than the ST-LSTM-based PredRNN-V2, although it remains behind PredRNN-V2 on pixel-level metrics. We expect that adapting PredRNN-V2’s decoupling loss to the ST-GRU will close part of this gap.

References

- Ballas, N., Yao, L., Pal, C., and Courville, A. (2016). Delving deeper into convolutional networks for learning video representations. In *International Conference on Learning Representations (ICLR)*.
- Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., et al. (2023). Improving image generation with better captions. OpenAI technical report, <https://cdn.openai.com/papers/dall-e-3.pdf>.
- Cai, H., Bai, C., Tai, Y.-W., and Tang, C.-K. (2018). Deep video generation, prediction and completion of human action sequences. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
- Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- Dong, J., Zhang, J., Xie, B., Zhang, J., Liu, C., and Cheng, W. (2025). ST-GRU: Spatiotemporal gated recurrent unit for video prediction. *Multimedia Systems*, 31(3).
- Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(2):107–116.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, volume 28.
- Wang, Y., Long, M., Wang, J., Gao, Z., and Yu, P. S. (2017). PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs. In *Advances in Neural Information Processing Systems*, pages 879–888.
- Wang, Y., Wu, H., Zhang, J., Gao, Z., Wang, J., Yu, P. S., and Long, M. (2021). PredRNN: A recurrent neural network for spatiotemporal predictive learning. *arXiv preprint arXiv:2103.09504*.
- Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.
- Wu, H. (2022). PredRNN: A recurrent neural network for spatiotemporal predictive learning (TPAMI 2022). <https://github.com/thuml/predrnn-pytorch>. Accessed: 2025-06-11.
- Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595.

A Model Architectures

A.1 Mathematical Comparison

$ \begin{aligned} g_t &= \tanh(W_{xg} * \mathcal{X}_t + W_{hg} * \mathcal{H}_{t-1}^l) \\ i_t &= \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1}^l) \\ f_t &= \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1}^l) \\ \mathcal{C}_t^l &= f_t \odot \mathcal{C}_{t-1}^l + i_t \odot g_t \\ \\ g'_t &= \tanh(W'_{xg} * \mathcal{X}_t + W'_{mg} * \mathcal{M}_t^{l-1}) \\ i'_t &= \sigma(W'_{xi} * \mathcal{X}_t + W'_{mi} * \mathcal{M}_t^{l-1}) \\ f'_t &= \sigma(W'_{xf} * \mathcal{X}_t + W'_{mf} * \mathcal{M}_t^{l-1}) \\ \mathcal{M}_t^l &= f'_t \odot \mathcal{M}_t^{l-1} + i'_t \odot g'_t \\ \\ o_t &= \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1}^l \\ &\quad + W_{co} * \mathcal{C}_t^l + W_{mo} * \mathcal{M}_t^l) \\ \mathcal{H}_t^l &= o_t \odot \tanh(W_{1 \times 1} * [\mathcal{C}_t^l, \mathcal{M}_t^l]) \end{aligned} $ (a) ST-LSTM	$ \begin{aligned} z_t &= \sigma(W_{xz} * \mathcal{X}_t + W_{hz} * \mathcal{H}_{t-1}^l + W_{mz} * \mathcal{M}_t^{l-1}) \\ r_t &= \sigma(W_{xr} * \mathcal{X}_t + W_{hr} * \mathcal{H}_{t-1}^l + W_{mr} * \mathcal{M}_t^{l-1}) \\ \widetilde{\mathcal{H}}_t^l &= \tanh(W_{xn} * \mathcal{X}_t + W_{rhn}(r_t \odot (W_{hn} * \mathcal{H}_{t-1}^l)) \\ &\quad + W_{mn} * \mathcal{M}_t^{l-1}) \\ \widetilde{\mathcal{H}}_t^l &= (1 - z_t) \odot \widetilde{\mathcal{H}}_t^l + z_t \odot \mathcal{H}_{t-1}^l \\ \\ z'_t &= \sigma(W'_{xz} * \mathcal{X}_t + W_{mz} * \mathcal{M}_t^{l-1}) \\ r'_t &= \sigma(W'_{xr} * \mathcal{X}_t + W_{mr} * \mathcal{M}_t^{l-1}) \\ \widetilde{\mathcal{M}}_t^l &= \tanh(W'_{xn} * \mathcal{X}_t + r'_t \odot (W_{mn} * \mathcal{M}_t^{l-1})) \\ \mathcal{M}_t^l = \widetilde{\mathcal{M}}_t^l &= (1 - z'_t) \odot \widetilde{\mathcal{M}}_t^l + z'_t \odot \mathcal{M}_t^{l-1} \\ \\ o_t &= \sigma(W_{ho} * \widetilde{\mathcal{H}}_t^l + W_{mo} * \widetilde{\mathcal{M}}_t^l) \\ \mathcal{H}_t^l &= o_t \odot \tanh(W_{1 \times 1} * [\widetilde{\mathcal{H}}_t^l, \widetilde{\mathcal{M}}_t^l]) \end{aligned} $ (b) ST-GRU
--	---

Figure 4: Side-by-side mathematical comparison of the ST-LSTM and ST-GRU building blocks.

A.2 Visual Comparison

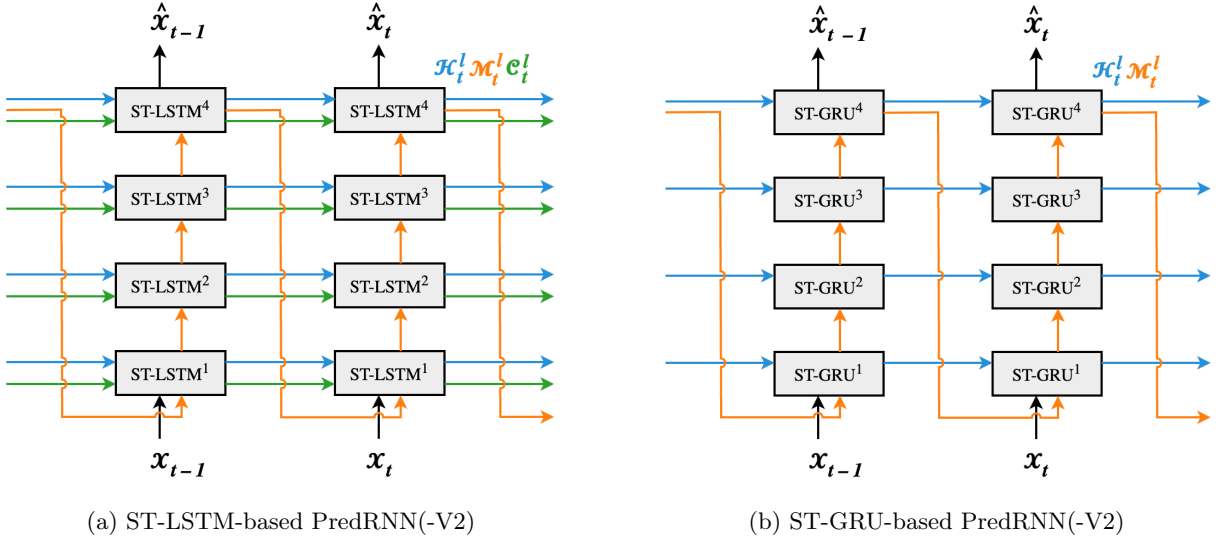


Figure 5: Side-by-side comparison of the ST-LSTM-based and ST-GRU-based PredRNN(-V2) architectures, with the flows of information shown in different colours. The ST-LSTM-based PredRNN(-V2) has two memory states: the temporal memory $\mathcal{C}_t^l$ flows horizontally from the previous time step to the current one (t), whereas the spatiotemporal memory $\mathcal{M}_t^l$ zigzags through the layers (l) of PredRNN(-V2), capturing short-term dynamics. Both memory states are used to update the hidden state $\mathcal{H}_t^l$. The ST-GRU-based PredRNN(-V2), in contrast, integrates the horizontal memory flow into the hidden state $\mathcal{H}_t^l$, while the spatiotemporal memory is still carried by $\mathcal{M}_t^l$.

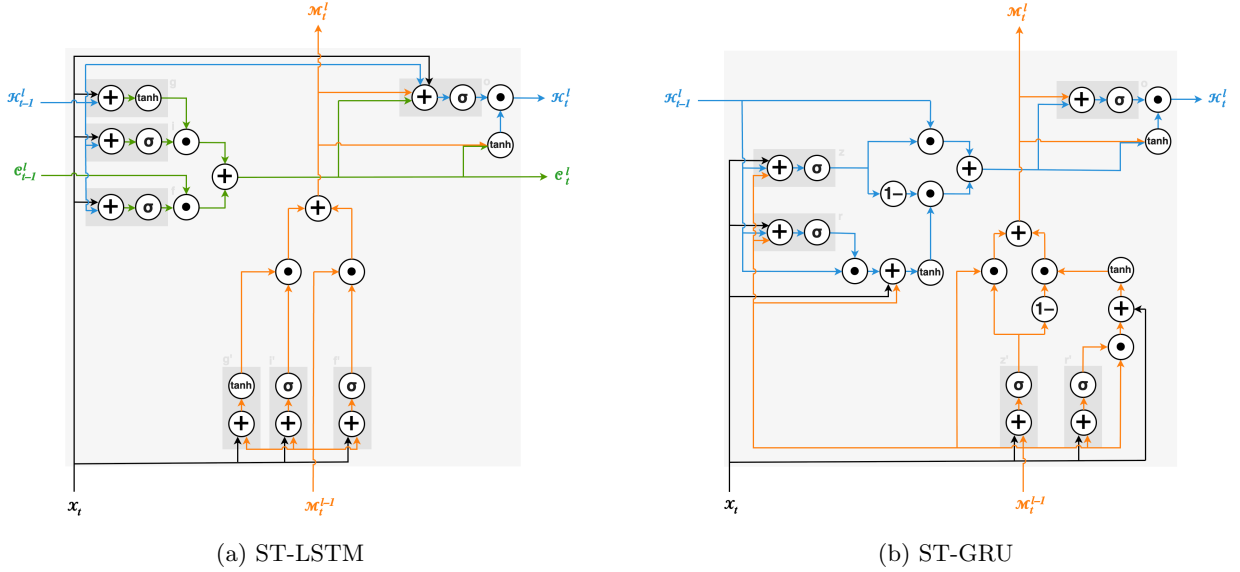


Figure 6: Side-by-side comparison of the ST-LSTM and ST-GRU building blocks, with the flows of information shown in different colours. The ST-LSTM has two memory states: the temporal memory $\mathcal{C}_t^l$ across time steps (t), and the spatiotemporal memory $\mathcal{M}_t^l$, which captures short-term dynamics across layers (l). Each memory state is modulated by three gates, i_t , f_t and g_t (respectively i_t' , f_t' and g_t'), and both contribute to the hidden state $\mathcal{H}_t^l$ via the output gate o_t . The ST-GRU, in contrast, integrates the horizontal memory flow into the hidden state $\mathcal{H}_t^l$, while the spatiotemporal memory is still carried by $\mathcal{M}_t^l$. Each of these states is modulated by two gates, z_t and r_t (respectively z_t' and r_t'), and they are combined via the output gate o_t . The reduction in both gates and memory states is what makes the ST-GRU simpler than the ST-LSTM.