SSS: Similarity-Based Scheduled Sampling for Video Prediction

Ryosuke Hata[1] and Yoshihisa Shinozawa[2]
[1]*Graduate School of Science and Technology, Keio University, Yokohama, Kanagawa, Japan*
[2]*Faculty of Science and Technology, Keio University, Yokohama, Kanagawa, Japan*

Keywords: Deep Learning, Video Prediction, Scheduled Sampling, Vision Transformer, Long Short-Term Memory.

Abstract: In video prediction tasks, numerous RNN-based models have demonstrated significant advancements. A well-established approach to enhancing these models during training is scheduled sampling. However, the adjustment of the probability parameter ε (scheduling) has not been adequately addressed, and current configurations are suboptimal for video prediction tasks. This issue arises because prior scheduling strategies depend solely on two factors: a function type and the total number of iterations, without considering the changes by motions, one of the most crucial features in videos. To address this gap, we propose similarity-based scheduled sampling, which accounts for the changes by motions. Specifically, we create difference frames between a given frame at a specific time step and another frame at a different time step for both the model's predicted output and the ground truth. We then assess the similarity of these difference frames at each iteration, to determine whether the changes by motions are properly trained and to incorporate it into the scheduling. Evaluation experiment shows that proposed method outperforms previous methods. Furthermore, an ablation study confirms the effectiveness of leveraging difference frames and demonstrates the significance of considering the changes by motions.

1 INTRODUCTION

Video prediction refers to the attempt of predicting and generating future videos from given past videos. This research area has attracted significant attention due to its wide-ranging applications in various fields such as anomaly detection, weather forecasting, and autonomous driving (Oprea et al., 2022).

In video prediction tasks, numerous RNN-based models have demonstrated significant advancements. These models generally employ an autoregressive structure where the decoder sequentially generates outputs. Scheduled sampling (Bengio et al., 2015), initially proposed in natural language processing, has been shown to be effective during training phase of models with autoregressive structures, demonstrating utility in video prediction as well (Wang et al., 2023). In the context of video prediction, Scheduled sampling operates by using ground truth videos with a probability of ε and the model's predicted videos with a probability of $1-\varepsilon$ at each decoding step during training phase.

As an improvement to scheduled sampling in natural language processing, (Liu et al., 2021a) propose scheduled sampling based on predicted translation probability (PTP), which is calculated as a measure of the model's confidence. (Liu et al., 2021b) also introduce a method that considers not only the training steps but also the decoding steps. Additionally, (Song et al., 2021) enhance scheduled sampling by incorporating an error correction mechanism.

Scheduled sampling has been widely adopted to train various video prediction models (Finn et al., 2016) (Wang et al., 2019) (Wang et al., 2017) (Wang et al., 2018) (Su et al., 2020) (Wang et al., 2023). Notably, (Wang et al., 2023) introduce reverse scheduled sampling, where the probability parameter ε adjusts inversely to the scheduled sampling.

A substantial gap exists between the characteristics of language and video, making it challenging to improve scheduled sampling for video prediction. The adjustment of the probability parameter ε (scheduling) has not been adequately addressed, and current configurations are suboptimal for video prediction tasks. Previous methods empirically select a pre-defined function (linear, exponential, sigmoid) before training and set function parameters based on the total number of iterations. This scheduling strategies depend solely on two factors: a function type and the total number of iterations, without considering the changes by motions, one of the most crucial features in videos. Therefore, to make scheduled sampling suitable for video prediction, it is necessary to determine whether it is properly trained the changes by

Hata, R. and Shinozawa, Y.
SSS: Similarity-Based Scheduled Sampling for Video Prediction.
DOI: 10.5220/0013297600003905

In *Proceedings of the 14th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2025)*, pages 122-129
ISBN: 978-989-758-730-6; ISSN: 2184-4313

motions and improve it so that it can be reflected in the scheduling. Consequently, to make scheduled sampling more suitable for video prediction, it is essential to determine whether the changes by motions are properly trained and to incorporate it into the scheduling.

To address this issue, we propose similarity-based scheduled sampling which accounts for the changes by motions. Specifically, we create difference frames between a given frame at a specific time step and another frame at a different time step for both the model's predicted output and the ground truth. We then assess the similarity of these difference frames at each iteration, to determine whether the changes by motions are properly trained and to incorporate it into the scheduling. We use perceptual hash to compute the similarity. Evaluation experiment demonstrate the superiority of the proposed method. Furthermore, we conduct an ablation study to validate the efficacy of using difference frames.

The main contributions of this paper are as follows:

- We propose scheduled sampling that utilizes the similarity calculated from difference frames obtained from the model's predicted output and ground truth. By incorporating into the scheduling whether the model can properly train the changes by motions, the settings are made suitable for video prediction. Evaluation experiment shows that proposed method outperforms previous methods.
- An ablation study confirms the effectiveness of leveraging difference frames and demonstrates the significance of considering the changes by motions.
- As our method improves the training approach with a simple algorithm, it can be widely and easily implemented in RNN-based models for video prediction.

2 RELATED WORK

2.1 Video Prediction Models

In the field of video prediction using deep learning, we can divide recent methods into three categories: recurrent neural network (RNN)-, convolutional neural network (CNN)-, and vision transformer (ViT)-based models.

RNN-based models have demonstrated notable success in video prediction. Convolutional LSTM Network (Shi et al., 2015) extends the LSTM network to handle videos by incorporating convolutional operations. CDNA (Finn et al., 2016) merges appearance information from previous frames with motion predicted by the model. PredRNN (Wang et al., 2017) propose spatiotemporal LSTM(ST-LSTM) unit that is designed to extract spatial and temporal representations simultaneously. PredRNN++ (Wang et al., 2018) addresses the vanishing gradient problem through gradient highway unit and designed causal LSTM to capture short-term dynamics. E3D-LSTM (Wang et al., 2019) supplements short-term motion information by incorporating 3D convolutions. Convolutional Tensor-Train LSTM (Su et al., 2020) realizes fully convolutional higher-order LSTM model capable of efficiently training spatio-temporal correlations by proposing convolutional tensor-train decomposition. SwinLSTM (Tang et al., 2023) integrates the memory cell of Convolutional LSTM Network with the Swin Transformer Block (Liu et al., 2021c) to capture spatial and temporal dependencies. Models with one and four memory cells are presented as SwinLSTM-B and SwinLSTM-D, respectively.

CNN-based models are characterized by their lightweight and simple structure. PredCNN (Xu et al., 2018) employs a hierarchical stacking of Cascade Multiplicative Units (CMUs) using only CNNs, thereby achieving a structure similar to that of RNN-based models while significantly reducing training time in comparison. SimVP (Gao et al., 2022) relies solely on CNNs and proposes a hierarchical structure consisting of units formed by multiple group convolutions.

VPTR (Ye and Bilodeau, 2022), known as a ViT-based model, achieves performance comparable to that of RNN-based models by refining its loss function and training method. Additionally, this model alleviates the computational complexity of self-attention by independently computing self-attention along local spatial and temporal dimensions.

2.2 Scheduled Sampling and Reverse Scheduled Sampling

Scheduled sampling is a training enhancement method for models that generate outputs autoregressively using a sequence-to-sequence structure, such as RNNs. In this approach, ground truth is used with probability of ε, and the model's predicted output is used with probability of $1-\varepsilon$ at each decoding step during the training phase. The probability parameter ε is progressively reduced as training progresses. In other words, at the beginning of training, there is a higher likelihood of using ground truth, whereas in

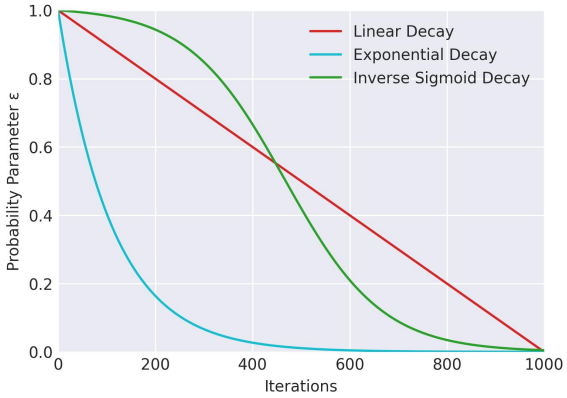

Figure 1: Scheduling strategy of scheduled sampling (Bengio et al., 2015). The probability parameter ε is progressively reduced as training progresses. This methods empirically select a pre-defined function (linear decay, exponential decay, inverse sigmoid decay) before training and set function parameters based on the total number of iterations.

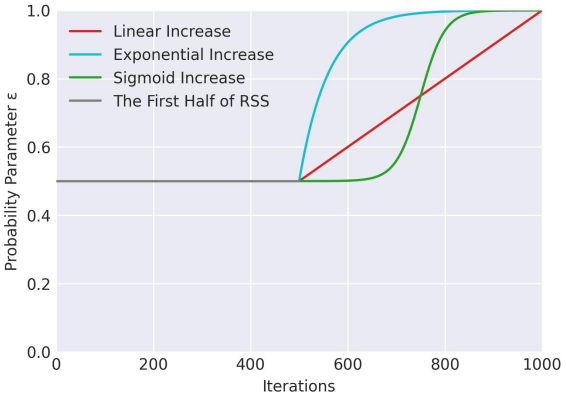

Figure 2: Scheduling strategy of reverse scheduled sampling (Wang et al., 2023). Like scheduled sampling, the probability parameter ε in reverse scheduled sampling is determined by one of three pre-defined functions but is adjusted inversely. The original paper suggests holding ε at 0.5 during the first half and applying the exponential function during the latter half. Additionally, this method can be combined with scheduled sampling to train a model.

later stages, the likelihood of using the model's predicted output increases. As shown in Figure 1, this probability parameter ε is determined by one of three pre-defined functions prior to training: Linear, Exponential, or Inverse Sigmoid.

Scheduled sampling has been widely adopted in training various video prediction models(Finn et al., 2016)(Wang et al., 2019)(Wang et al., 2017)(Wang et al., 2018)(Su et al., 2020)(Wang et al., 2023). (Wang et al., 2023) introduces reverse scheduled sampling to force the model to learn more about long-term dynamics. In reverse scheduled sampling, the probability parameter ε adjusts inversely to the scheduled sampling at the encoding step. Like scheduled sampling, the probability parameter ε in reverse scheduled sampling is determined by one of three pre-defined functions but is modified to increase over time.

As shown in Figure 2, the original paper suggests holding ε at 0.5 during the first half of training and applying the selected function during the latter half, with the exponential function gave the highest reported accuracy. Additionally, it is feasible to combine reverse scheduled sampling in the encoding step with scheduled sampling in the decoding step to train a model.

Previous methods empirically select a pre-defined function (linear, exponential, sigmoid) and set function parameters according to the total number of iterations. Hence, the scheduling decision process does not account for the changes by motions, which is one of the most crucial features of videos and is not suitable for video prediction.

2.3 Perceptual Hash Algorithm

The Perceptual Hash (pHash) algorithm computes hash values by extracting and leveraging features from images. Unlike neural networks, which require extensive training and large datasets, pHash algorithm are based on various techniques that do not depend on such resources. Among these, the calculation based on the Discrete Cosine Transform (DCT) is widely adopted due to its robustness (Du et al., 2020).

The hash calculation process begins by resizing the target image and converting it to grayscale. The image is then processed using DCT and low-frequency components are extracted sequentially. These components are binarized based on their median value to obtain the hash value. By comparing the hash values generated in this way, the similarity between different images can be assessed. The hash values are compared using the hamming distance, producing smaller values for similar images and larger values for dissimilar ones.

The hash length can be set to any arbitrary value, although it is commonly set to 64 (8×8). Increasing the hash length allows for the inclusion of more high-frequency components, enabling more detailed image comparisons.

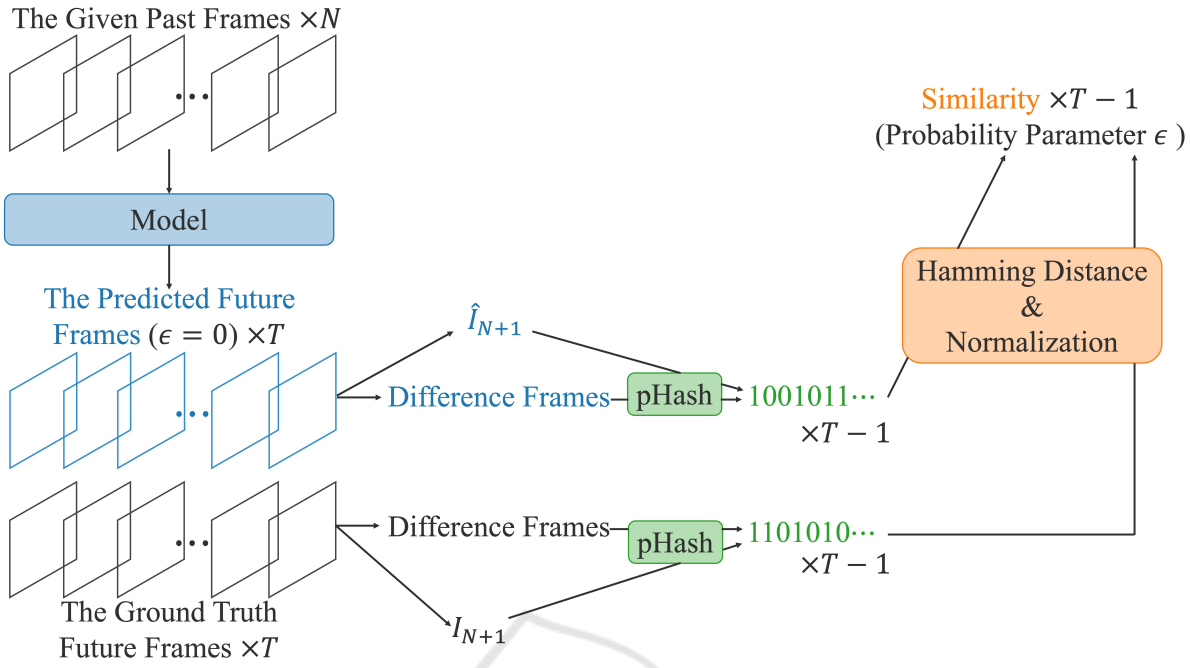

Figure 3: Overview of the similarity-based scheduled sampling. First, generate the predicted future frames using only predicted future frame at all decoding steps (equivalent to setting ε = 0.0). Difference frames are then created from both the predicted future frames and the ground truth future frames. Next, apply pHash to the difference frames to compute hash values. Then, calculate the hamming distance between the hash values of the difference frames at corresponding time steps and normalize to a range of 0.0 to 1.0 by dividing by the hash length. This value is adopted as the probability parameter ε for training.

3 METHOD

3.1 Problem Definition

In this paper, as with general video prediction, we predict and generate a video (the predicted future frames) consisting of the following T frames from a video (the given past frames) consisting of N frames. The video prediction task can be formulated as follows:

$$\theta' = \underset{\theta}{\operatorname{argmin}} \mathcal{L}((I_{N+1}, \ldots, I_{N+T}), f_\theta(I_1, \ldots, I_N)) \quad (1)$$

where $I_1, \ldots, I_N$ denotes the given past frames, $I_{N+1}, \ldots, I_{N+T}$ denotes the ground truth future frames, θ denotes learnable model parameters, $\mathcal{L}$ denotes the loss function, f denotes the model. Furthermore, we define the video that corresponds to the ground truth future frames as the predicted future frames $(\hat{I}_{N+1}, \ldots, \hat{I}_{N+T})$, and the temporal sequence of frames in the videos as time steps.

Video prediction using RNN-based encoder-decoder models involves inputting the given past frames into the encoder and generate the predicted future frames by autoregressively using the decoder. it is desirable for the predicted future frames to be close to the ground truth future frames.

3.2 Similarity-Based Scheduled Sampling

Scheduled sampling and reverse scheduled sampling have limitations when applied to video prediction, as described in 2.2. This issue is likely due to the absence of comparisons that account for the changes by motions. Therefore, to make scheduled sampling more suitable for video prediction, it is essential to determine whether the changes by motions are properly trained and to incorporate it into the scheduling.

To address this, we propose similarity-based scheduled sampling that utilizes the similarity calculated from the difference frames between the predicted future frames and the ground truth future frames and incorporates this similarity into the scheduling at each iteration.

As shown in Figure 3, we first generate the predicted future frames using only predicted future frame at all decoding steps (equivalent to setting ε = 0.0). Difference frames are then created from both the

predicted future frames and the ground truth future frames, as described in 3.3. Next, we apply pHash to the difference frames to compute hash values and calculate the hamming distance between the hash values of the difference frames at corresponding time steps in the predicted and ground truth future frames. Then, we normalize to a range of 0.0 to 1.0 by dividing by the hash length. The calculated value actually represents dissimilarity. However, in this paper, it is appropriate to adopt this value directly as probability parameter ε for training. Therefore we treat it as a similarity.

When the similarity approaches 1, the predicted future frames and the ground truth future frames are dissimilar, indicating that the model has not effectively trained the changes by motions. In this case, we should increase the probability parameter ε of using the ground truth future frames. Conversely, as the similarity approaches 0.0, the predicted and ground truth future frames are similar, suggesting the model has successfully trained the changes by motions, and thus the probability parameter ε of using the predicted future frames should be increased. In scheduled sampling, the ground truth is used with probability of ε, and model's predicted output is used with a probability of 1-ε. Therefore, the similarity is directly adopted as the probability ε.

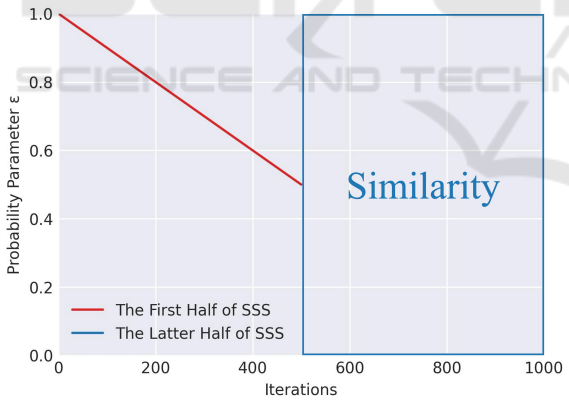

Figure 4: Scheduling strategy of similarity-based scheduled sampling. We propose similarity-based scheduled sampling that utilizes the similarity calculated from the difference frames between the predicted future frames and the ground truth future frames and incorporates this similarity into the scheduling at each iteration. The first half of the training uses scheduled sampling, while the latter half utilizes similarity-based scheduled sampling.

As shown in Figure 4, the first half of the training uses scheduled sampling, while the latter half utilizes similarity-based scheduled sampling. This design is based on the necessity to progress training to a certain extent to enhance the quality of the predicted future frames. In the initial scheduled sampling phase, we applied a linear function to adjust ε from 1.0 to 0.5. The hash length set to 1024 (32×32), to enable a more detailed comparison of the images by using a longer than normal hash length.

3.3 Difference Frames

The proposed method creates difference frames from both the predicted and ground truth future frames, which are then processed individually. Specifically, difference frames are created with reference to the frame at $N+1$ (I_{N+1}). However, only the frame at $N+1$ is used without any processing. This is because it is the reference frame and difficult to compare with itself. And the difference frame at time T is not created. Because the last frame of the predicted future frames (I_{N+T}) is not input to the model with an autoregressive structure. The difference frames of the predicted future frames (PD_{N+t}) can be computed as follows:

$$PD_{N+t} = \begin{cases} \hat{I}_{N+t} - \hat{I}_{N+1} & \text{if } 2 \leq t \leq T-1, \\ \hat{I}_{N+1} & \text{if } t = 1, \\ \text{None} & \text{if } t = T. \end{cases} \quad (2)$$

Similarly, the difference frames of the ground truth future frames (GD_{N+t}) can be computed as follows:

$$GD_{N+t} = \begin{cases} I_{N+t} - I_{N+1} & \text{if } 2 \leq t \leq T-1, \\ I_{N+1} & \text{if } t = 1, \\ \text{None} & \text{if } t = T. \end{cases} \quad (3)$$

By applying the pHash to these created difference frames, hash values are calculated. The similarity between the corresponding difference frames in the time steps is then determined by comparing these hash values.

4 EXPERIMENTS

4.1 Dataset

To train and validate the proposed method, we used the Autonomous Driving Dataset (A2D2 dataset) (Geyer et al., 2020). This dataset consists of 1,920×1,280 color videos at 30fps and includes semantic segmentation images, point cloud labels and 3D bounding boxes. In this paper, we used videos which was recorded using a camera attached to the center of the front of an automobile in Gaimersheim, southern Germany. We input the given past frames of 1.5 seconds into the model and generate the predicted future frames of 1.6 seconds.

Table 1: Results on the A2D2 dataset.

Model	MSE↓
SwinLSTM-D (Tang et al., 2023)	749
SwinLSTM-D + SS (Bengio et al., 2015)	729
SwinLSTM-D + SS&RSS (Bengio et al., 2015)(Wang et al., 2023)	812
SwinLSTM-D + SSS (ours)	**691**

4.2 Implementation

We selected SwinLSTM-D, a model that demonstrates high accuracy among RNN-based models. We trained the model according to the hyperparameter settings published in the original paper and on the official GitHub repository. We resized the A2D2 dataset to 320×224 and normalized the RGB values to the range of 0.0 to 1.0. Then, we skipped three frames ($N = T = 12$).

4.3 Evaluation

We evaluated the proposed method using Mean Squared Error (MSE) on 904 videos from the A2D2 dataset that prepared without overlapping with those used in training. MSE is a pixel-wise measure of the difference between the predicted future frames and the ground truth future frames. It serves as a metric to quantify how closely the predicted future frame with the ground truth future frame. A lower MSE value indicates a more desirable model's output.

Table 1 shows a comparison of the results when scheduled sampling is applied to SwinLSTM-D and when both reverse scheduled sampling and scheduled sampling are combined. For the SwinLSTM-D listed at the top, the encoder was trained using the ground truth future frame, and the decoder was trained using the predicted future frame at each time step (Sutskever et al., 2014). Scheduled sampling is abbreviated as SS and reverse scheduled sampling is abbreviated as RSS.

In the implementation of SS, we employed a linear function to vary the probability parameter ε from 1.0 to 0.0. For reverse scheduled sampling in the SS&RSS approach, an exponential function was used: the probability parameter ε was set to 0.5 during the first half of training and then increased from 0.5 to 1.0 in the latter half. This choice was based on findings from the original paper, which reported that an exponential function achieved the highest accuracy. For comparison, scheduled sampling in SS&RSS also used a linear function to adjust probability parameter ε from 1.0 to 0.0. In addition to SS&RSS, we also evaluated the case of applying only RSS and the case of adjusting the hyperparameters of the RSS exponential function in several ways. The best of these values is shown in Table 1.

As shown in Table 1, the proposed method achieved the highest accuracy.

4.4 Ablation Study

To validate the efficacy of utilizing difference frames, we compare similarity-based scheduled sampling, which utilizes the similarity of vanilla frames (without any processing) instead of the difference frames. In other words, this means utilizing the similarity obtained by directly comparing corresponding vanilla frames at all time steps, which is the same as the comparison between $\hat{I}_{N+1}$ and I_{N+1}in the proposed method.

The vanilla frames of the predicted future frames (PV_{N+t}) can be represented as follows:

$$PV_{N+t} = \begin{cases} \hat{I}_{N+t} & \text{if } 1 \leq t \leq T-1, \\ \text{None} & \text{otherwise.} \end{cases} \tag{4}$$

Similarly, the vanilla frames of the ground truth future frames (GV_{N+t}) can be represented as follows:

$$GV_{N+t} = \begin{cases} I_{N+t} & \text{if } 1 \leq t \leq T-1, \\ \text{None} & \text{otherwise.} \end{cases} \tag{5}$$

Table 2 shows the comparison results.

Table 2: Ablation study results on the A2D2 dataset.

Model	MSE↓
SwinLSTM-D + SSS (vanilla frames)	723
SwinLSTM-D + SSS (ours)	**691**

As shown in Table 2, the accuracy based on similarity derived from difference frames surpasses that of the vanilla frames approach by approximately 4.4%.

4.5 Discussion

The results of applying scheduled sampling to SwinLSTM-D and combining reverse scheduled and scheduled sampling show that the proposed method outperforms previous methods. This improvement can be attributed to the utilization of similarity between difference frames of the predicted future frames and the ground truth future frames, allowing

for a scheduling strategy that is well-suited to video prediction.

An ablation study comparing similarity based on vanilla frames (without any processing) reveals that determining the probability parameter ε while accounting for the changes by motions is crucial, and further confirms effectiveness of leveraging difference frames.

Limitations of the proposed method include the higher computational cost of calculating similarity compared to previous methods, and the difficulty in determining the optimal settings for the hash length and the extent to which training should proceed to improve the quality of the model's output.

5 CONCLUSION

In this paper, we have introduced similarity-based scheduled sampling that utilizes the similarity calculated from difference frames. This approach addresses the challenge of setting a scheduling strategy suited to video prediction tasks. The proposed method outperforms previous methods. Furthermore, an ablation study demonstrates the importance of determining the probability parameter ε considering the changes by motions.

We plan to work on exploring alternative methods other than difference frames and reducing computational costs.

REFERENCES

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1171–1179.

Du, L., Ho, A. T., and Cong, R. (2020). Perceptual hashing for image authentication: A survey. *Signal Processing: Image Communication*, 81:115713.

Finn, C., Goodfellow, I., and Levine, S. (2016). Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 64–72.

Gao, Z., Tan, C., Wu, L., and Li, S. Z. (2022). Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (ICCV)*, pages 3160–3170.

Geyer, J., Kassahun, Y., Mahmudi, M., Ricou, X., Durgesh, R., Chung, A. S., Hauswald, L., Pham, V. H., Mühlegg, M., Dorn, S., et al. (2020). A2d2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320*.

Liu, Y., Meng, F., Chen, Y., Xu, J., and Zhou, J. (2021a). Confidence-aware scheduled sampling for neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2327–2337, Online. Association for Computational Linguistics.

Liu, Y., Meng, F., Chen, Y., Xu, J., and Zhou, J. (2021b). Scheduled sampling based on decoding steps for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3285–3296, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021c). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002.

Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J. A., Orts-Escolano, S., Garcia-Rodriguez, J., and Argyros, A. (2022). A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2806–2826.

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 (NeurIPS)*, pages 802–810.

Song, K., Tan, X., and Lu, J. (2021). Neural machine translation with error correction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Su, J., Byeon, W., Kossaifi, J., Huang, F., Jan, K., and Anandkumar, A. (2020). Convolutional tensor-train lstm for spatio-temporal learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13714–13726.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3104–3112.

Tang, S., Li, C., Zhang, P., and Tang, R. (2023). Swinlstm: Improving spatiotemporal prediction accuracy using swin transformer and lstm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13424–13433.

Wang, Y., Gao, Z., Long, M., Wang, J., and Philip, S. Y. (2018). Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 5123–5132. PMLR.

Wang, Y., Jiang, L., Yang, M.-H., Li, L.-J., Long, M., and Fei-Fei, L. (2019). Eidetic 3d LSTM: A model for video prediction and beyond. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

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 (NeurIPS), pages 879–888.

Wang, Y., Wu, H., Zhang, J., Gao, Z., Wang, J., Yu, P. S., and Long, M. (2023). Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2208–2225.

Xu, Z., Wang, Y., Long, M., Wang, J., and KLiss, M. (2018). Predcnn: Predictive learning with cascade convolutions. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 2940–2947.

Ye, X. and Bilodeau, G.-A. (2022). Vptr: Efficient transformers for video prediction. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 3492–3499.