A Hierarchical Variational Neural Uncertainty Model for Stochastic Video Prediction

Moitreya Chatterjee[1] Narendra Ahuja[1] Anoop Cherian[2]
[1]University of Illinois at Urbana-Champaign Champaign, IL 61820, USA
[2]Mitsubishi Electric Research Laboratories, Cambridge, MA 02139
metro.smiles@gmail.com n-ahuja@illinois.edu cherian@merl.com

1. Appendix

In this supplementary document, we provide several additional results that further bring out the benefits of our approach. We start with a visualization of the predicted uncertainties per-frame, across different datasets in Section 2. Followed by a brief review of alternative formulations to Neural Uncertainty Quantifier (NUQ) that one may come up with in Section 3 and demonstrate that our proposed formulation performs the best among these alternatives. In Section 4, we discuss the details of the architectural design of the NUQ framework. Next, in Section 5, we present the derivations to Eq. 9 and 11 in the paper. We then quantitatively evaluate the diversity of the generated futures in Section. 6. We present several qualitative results thereafter, which also showcases the diversity in frame generation of NUQ. We list these items below:

1. Uncertainty Visualization
2. Alternative formulations of NUQ.
3. Architectural Details.
4. Derivations for Eq. 9 and 11.
5. Quantitatively evaluate the diversity of the generated samples.
6. Qualitative Results on all datasets

2. Uncertainty Visualizations

Figure 1 visualizes the scaled uncertainty values against the visual frames, across the SMMNIST and BAIR Push datasets, each trained with 2000 samples. See the caption of the figure for more details.

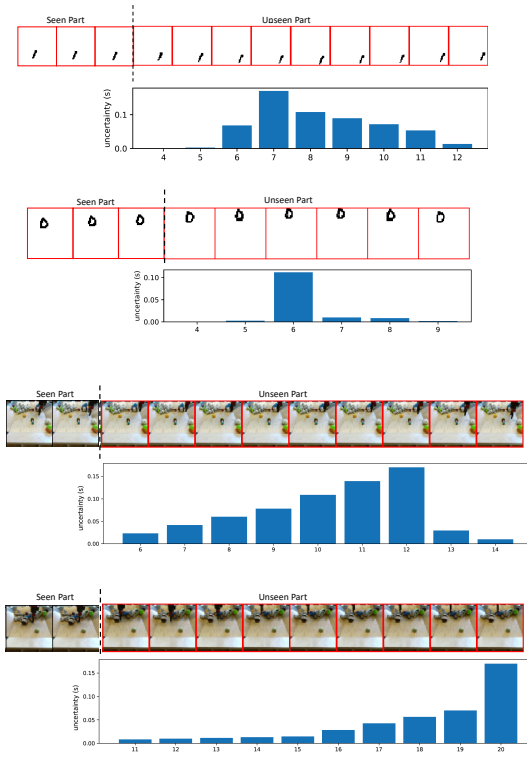


Figure 1. Top two rows: Evolution of scaled uncertainty on the SMMNIST dataset (with NUQ trained with 2,000 training samples on the SMMNIST Dataset) against time-steps. The plot shows the increase in uncertainty co-occurs with the bounce of the digit against the boundary. Bottom two rows: Scaled Uncertainty against time-steps on BAIR Push (with NUQ trained with 2000 training samples on the BAIR Push Dataset), showing that uncertainty co-occurs with the occlusion of the robot arm.

3. Alternative Formulations

Is NUQ the best formulation that one could have for quantifying uncertainty within a stochastic prediction model? In this section, we propose several alternatives and empirically evaluate them against the results we obtained using our formulation of NUQ, as an answer to this interesting research question.

3.1. Alternatives

Alternative Priors: In this variant, we replace our empirical gamma hyperprior, $p(s_t)$, with a half-normal distribution with location set to 0 and scale set to 1, and in another variant we use the $\text{Uniform}(0, 1)$ distribution as the hyperprior.

Using Mahalanobis Distance: In this variant, we use the frame decoder, $p_\theta(\cdot)$, to produce a diagonal covariance instead of producing the parameters of the gamma prior. Specifically, the output layer of the decoder now predicts both the future frame and the diagonal elements of $\mathbf{\Sigma}_t^{\boldsymbol{z}}$, where we assume $\mathbf{\Sigma}_t^{\boldsymbol{z}}$ is $n \times n$, decoded frames are size $d \times d$, and $D = d^2$. Generating an estimate of the output covariance matrix thus implies predicting the D terms along the diagonal of this matrix. We then reshape these D terms to $d \times d$ to match with the pixel resolution of the frames. We then use this uncertainty (precision) to weigh the MSE at a pixel level (instead of the precision b_t). This uncertainty is visualized for a sequence in Figure 2.

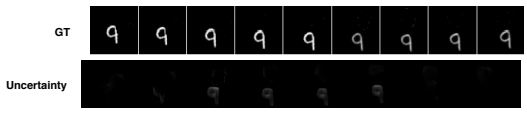


Figure 2. Visualization of pixel-wise uncertainty obtained by estimating the variance of the output, directly from the decoder for the SMMNIST Dataset, trained with 2,000 training samples.

Directly Estimate precision from latent prior: In this formulation, we repurpose the variance encoder ζ_λ, to emit the variance to the MSE, b_t, directly. This is in contrast with the architecture of the variance encoder in NUQ, where it is used to estimate the sufficient statistics of the truncated normal distribution, $\alpha_t^{\tilde{s}}$ and $\beta_t^{\tilde{s}}$. We essentially replace the final hidden layer of the network with a single neuron with sigmoid activation in order to realize this setting.

3.2. Alternatives – Results

In Table 1, we provide comparisons of the above alternative formulations on the SMMNIST dataset, trained with 2,000 training samples. From the first row, we see that the Uniform$[0, 1]$ distribution variant under-performs compared to using the gamma distribution as a hyper-prior, as in NUQ. We surmise that this is due to these distributions being more spread out over the probability space, as a result of which they often sample s_t's which do not match the true underlying distribution. This results in the MSE term in the loss function, being overly weighed when it should not be and vice-versa. Results for our other alternative, to compute the Mahalanobis-type precision matrix directly from the frame decoder, is provided in the second row in Table 1; its performance is similar to the other alternatives. We also attempted to directly estimate s_t from the variance of the latent space prior $\mathbf{\Sigma}_t^{\boldsymbol{z}}$. The results for this setting are shown in the third row in Table 1. However, this setting performs poorly suggesting that a deterministic mapping of the $\mathbf{\Sigma}_t^{\boldsymbol{z}}$ to s_t is not ideal, perhaps because the difference in the uncertainty distribution in the latent space and in the output is not accurately modeled this way. Overall, the results in the table clearly show that our proposed formulation of the model yields the best empirical performance, nonetheless some other formulations to our model seem promising.

4. Architectural Details

In this section, we elaborate on some of the architectural design choices that we made while implementing NUQ. Our primary objective while designing the architectural framework of NUQ was to ensure that our network's generation capacity remained similar to the state-of-the-art baselines, such as Denton and Fergus [1], such that all gains obtained by our framework, could be attributed to modeling the prediction uncertainty.

4.1. Frame Encoder

Our frame encoder consists of a hierarchical stack of 2d-convolution filters. For 48×48 inputs, we design a 4-layer network. The first layer consists of 64, 4×4 2d-convolutional filters with stride 2 and padding 1, which are followed by 2d-BatchNorm and LeakyReLU non-linearity. In every subsequent layers, we keep doubling the number of filters. For 64×64 inputs, we adapt this network to make it a 5-layer one.

4.2. LSTMs

All LSTM modules in our framework, including the sequence discriminator, have a single hidden layer with 256-d hidden states, except for the LSTM in the frame decoder $p_\theta(\cdot)$, which has 2 hidden layers, each of 256-d.

4.3. Frame Decoder

We design the frame decoder in congruence with the frame encoder, so as to permit skip connections between them, in a U-Net style network [3]. Therefore, our frame decoder obeys a similar architecture akin to the frame encoder, except the 2d-convolution filters are now replaced with 2d-deconvolution filters and the number of filters in each layer is doubled (in order to accommodate the skip connection).

4.4. Variance Encoder

Our variance encoder, $\zeta_\lambda(\cdot)$, is a 2-layer multi-layer perceptron, with LeakyReLU activations, which ultimately produces the sufficient statistics of the truncated normal distribution governing the posterior in the latent space.

Table 1. Best SSIM, PSNR, and LPIPS scores on the SMMNIST test set after @1, @5, and @Convergence (C) (upto 150) epochs of training with alternative formulations of our model using 2,000 training samples. [Key: Best results in **bold**].

Dataset: SMMNIST	**SSIM** ↑			**PSNR** ↑			**LPIPS** ↓		
	@1	@5	@C	@1	@5	@C	@1	@5	@C
$p(s_t) \sim$ Uniform$[0, 1]$	0.8173	0.8374	0.8523	17.6	17.95	18.06	0.3442	0.3038	0.198
Estimate b_t from the decoder $p_\theta(\cdot)$	0.7627	0.7628	0.7828	17.54	17.55	17.55	0.3463	0.3259	0.2225
Estimate b_t w/o variance encoder-decoder	0.7450	0.7454	0.7648	16.22	16.53	16.78	0.3469	0.3263	0.2328
NUQ (Ours)	**0.8686**	**0.8638**	**0.8948**	**17.76**	**18.13**	**18.14**	**0.3087**	**0.2836**	**0.1803**

5. Derivations

In this section, we present the derivations of Eq. 9 and Eq. 11 in the paper. We derive Eq. 9, along the lines of the variational lower bound derivation in Kingma and Welling [2]:

$$\begin{aligned}\log p(\boldsymbol{x}_t|b_t, \boldsymbol{x}_{1:t-1}) &= \log p(\boldsymbol{x}_t|b_t, \boldsymbol{x}_{1:t-1}). \int_{\boldsymbol{z}_t} q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t}) d\boldsymbol{z}_t \\ &= \int_{\boldsymbol{z}_t} q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t}) \log \frac{p(\boldsymbol{x}_t, \boldsymbol{z}_t|b_t, \boldsymbol{x}_{1:t-1})}{p(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})} d\boldsymbol{z}_t \\ &= \int_{\boldsymbol{z}_t} q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t}) \log \frac{p(\boldsymbol{x}_t, \boldsymbol{z}_t|b_t, \boldsymbol{x}_{1:t-1})}{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})} d\boldsymbol{z}_t \\ &\quad + \int_{\boldsymbol{z}_t} q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t}) \log \frac{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})}{p(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})} d\boldsymbol{z}_t\end{aligned} \tag{1}$$

The second term in the above equation is essentially a KL-Divergence, which is non-negative. We therefore have:

$$\begin{aligned}\log p(\boldsymbol{x}_t|b_t, \boldsymbol{x}_{1:t-1}) &\geq \int_{\boldsymbol{z}_t} q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t}) \log \frac{p(\boldsymbol{x}_t, \boldsymbol{z}_t|b_t, \boldsymbol{x}_{1:t-1})}{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})} d\boldsymbol{z}_t \\ &= \int_{\boldsymbol{z}_t} q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t}) \log \frac{p(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_t, b_t) p(\boldsymbol{z}_t|\boldsymbol{x}_{1:t-1})}{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})} d\boldsymbol{z}_t\end{aligned} \tag{2}$$

This yields Eq. 9, when the expression inside the log is split into two, with the first term amounting to the expectation term in Eq. 9, while the second one resulting in the KL-term.

Our NUQ framework is a essentially, a hierarchical variational encoder-decoder network, where the second level of the hierarchy is described by Eq. 11. Derivation for Eq. 11, thus proceeds analogously to Eq. 9, as follows:

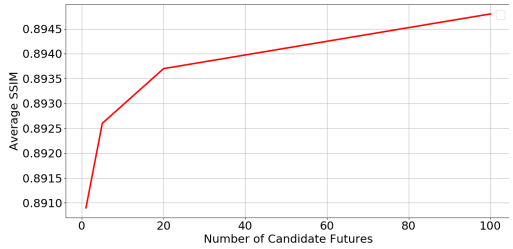
$$\begin{aligned}\log p(b_t|\boldsymbol{x}_{1:t-1}) &= \log p(b_t|\boldsymbol{x}_{1:t-1}). \int_{s_t} q_\lambda(s_t|\boldsymbol{x}_{1:t-1}) ds_t \\ &= \int_{s_t} q_\lambda(s_t|\boldsymbol{x}_{1:t-1}) \log \frac{p(b_t, s_t|\boldsymbol{x}_{1:t-1})}{p(s_t|b_t, \boldsymbol{x}_{1:t-1})} ds_t \\ &= \int_{s_t} q_\phi(s_t|\boldsymbol{x}_{1:t-1}) \log \frac{p(b_t, s_t|\boldsymbol{x}_{1:t-1})}{q_\lambda(s_t|\boldsymbol{x}_{1:t-1})} ds_t \\ &\quad + \int_{s_t} q_\lambda(s_t|\boldsymbol{x}_{1:t-1}) \log \frac{q_\lambda(s_t|\boldsymbol{x}_{1:t-1})}{p(s_t|b_t, \boldsymbol{x}_{1:t-1})} ds_t\end{aligned} \tag{3}$$

Like before, the second term in the above equation is essentially a KL-Divergence, which is non-negative. We therefore have:

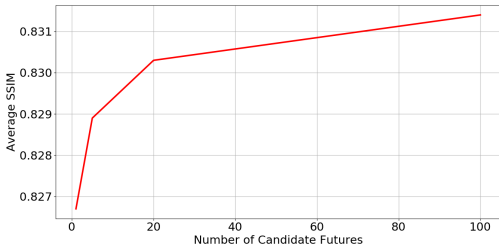
$$\begin{aligned}\log p(b_t|\boldsymbol{x}_{1:t-1}) &\geq \int_{s_t} q_\lambda(s_t|\boldsymbol{x}_{1:t-1}) \log \frac{p(b_t, s_t|\boldsymbol{x}_{1:t-1})}{q_\lambda(s_t|\boldsymbol{x}_{1:t-1})} ds_t \\ &= \int_{s_t} q_\lambda(s_t|\boldsymbol{x}_{1:t-1}) \log \frac{p(b_t|\boldsymbol{x}_{1:t-1}, s_t) p(s_t)}{q_\lambda(s_t|\boldsymbol{x}_{1:t-1})} ds_t\end{aligned} \tag{4}$$

When the expression inside the log is split into two, the first term results in the expectation term in Eq. 11, while the second one amounts to the KL-term.

6. Quantitative Evaluation of Diversity



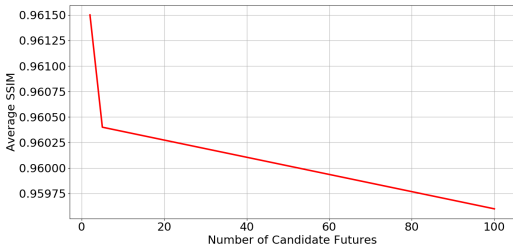
(a) SMMNIST - SSIM



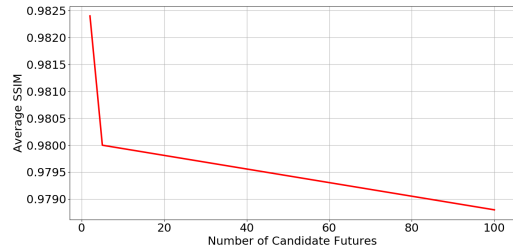
(b) BAIR Push - SSIM

Figure 3. Diversity in Generated Futures: Evaluation of diversity in the generation using SSIM on: (a) SMMNIST, (b) BAIR-Push for increasing number of candidate futures, computed by comparing against the ground truth (higher the better). We used 2000 samples for training NUQ for both datasets.

In order to analyze the extent of diversity in the generated frames of our model, we first resort to quantitative evaluation. In Figures 3(a), 3(b), we plot the average SSIM scores (over time steps) against the number of generated future candidates



(a) SMMNIST-Intra



(b) BAIR-Intra

Figure 4. Diversity in Generated Futures: Evaluation of diversity in the generation: (a,b) shows diversity in the generated futures by comparing intra-SSIM distances between all the futures, at a given time step, and computing the average (lower the better), for SMMNIST and BAIR Push respectively. For each of these datasets NUQ was trained with 2000 samples.

per time-step for each of the three datasets. For purposes of these plots, the SSIM is computed between the generated samples and the ground-truth. The monotonically increasing curve, in these figures, suggests straightforwardly, that sampling more future per time step helps in better generation, resulting from the synthesis of more accurate samples - indicating the model's diversity. In Figures 4(a), 4(b), we plot the average SSIM and PSNR scores between every pair of candidates generated in each time step, against the number of futures. These plots decrease monotonically, suggesting greater difference (i.e. diversity) between the generated frames as the number of sampled futures goes up. See the figure caption for more details.

7. Qualitative Results

We next present visualizations of frames generated using NUQ vis-á-vis competing baselines, on the SMMNIST, BAIR push, and KTH Action datasets. Also shown are diverse frame generations by NUQ for each of these datasets.

The results in Figures 5, 8, 9, 10, 11, 12, 13, 14, 15, 16 show qualitative generation results on the SMMNIST dataset, trained with 2000 samples. Besides the superior quality of the results generated by our method, we note that for some cases such as in Figures 11, 12, 13 the prediction of the baseline method simply disappears. We surmise that this

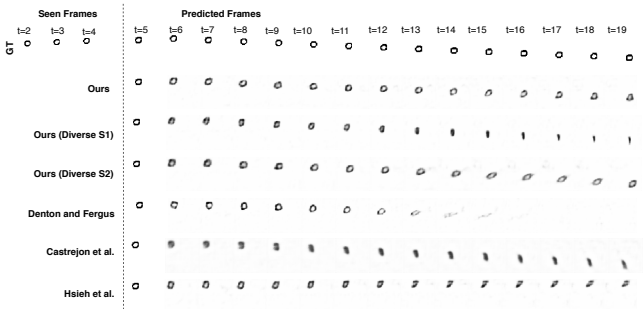


Figure 5. Visualization of generations by our method versus competing baselines on the SMMNIST Dataset, trained with 2,000 training samples. Further, diverse generations by our method are also shown.

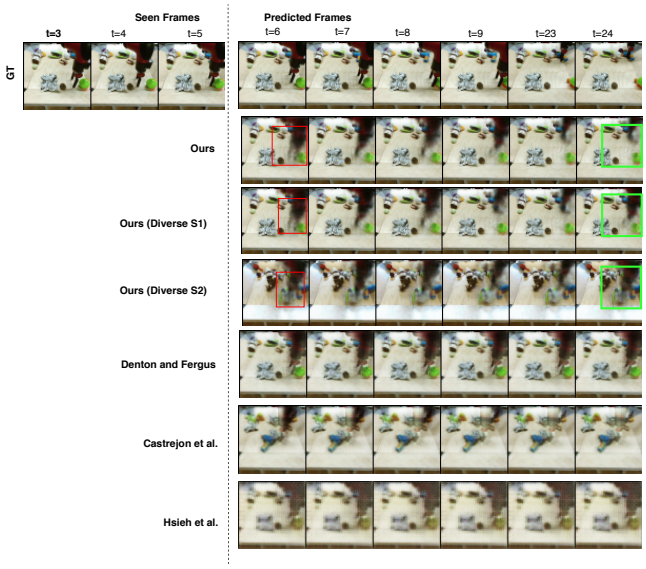


Figure 6. Visualization of generations by our method versus competing baselines on the BAIR Robot Push Dataset, trained with 2,000 training samples. Further, diverse generations by our method are also shown. High motion regions are indicated by a red bounding box, while spatial regions exhibiting high diversity are shown by a green bounding box.

is due to their inability to learn the motion dynamics of the digit well, in uncertain environments. In particular, if the motion is not aptly learnt, then the model often gets penalized heavily for inaccurately placing a digit (via the MSE loss), since this results in a high pixel-wise error. In such a scenario, a model might prefer to not display the digit at all. However, high stochasticity in the data may not suit this well and as a result might hurt the generalization. By intelligently down-weighting the MSE, we circumvent this problem.

We also see in Figures 6, 17, 21, 18, 22, 23, 24, 25, 19, 20 the performance of different competing methods versus NUQ on the BAIR push dataset, trained with 2000 samples. The figures reveal that our method captures the motion

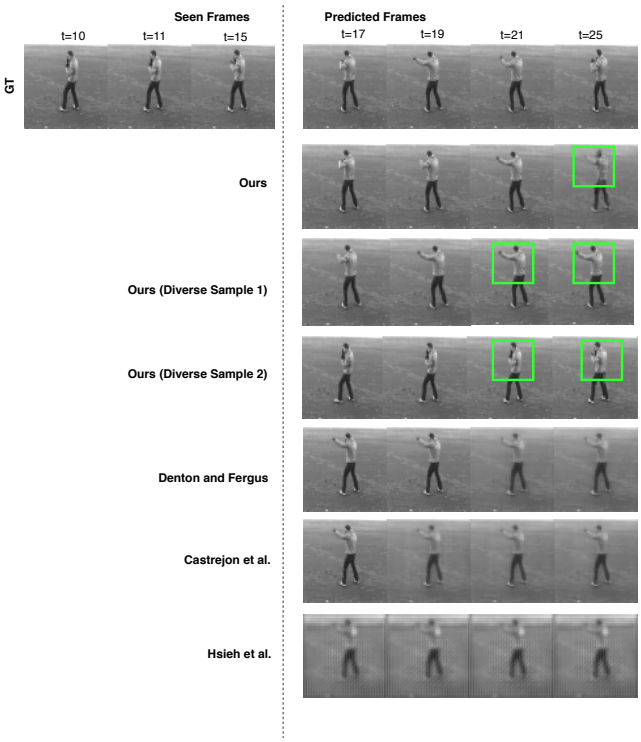


Figure 7. Visualization of generations by our method versus competing baselines on the KTH Action Dataset, trained with the full training data of 1,911 training samples. Further, diverse generations by our method are also shown. Spatial regions exhibiting high diversity are shown by a green bounding box.

of the robot arm, reasonably well, compared to competing methods.

Figures 7, 26, 27 present sample generation results by our method versus competing baselines on the KTH Action dataset. From the figures, we see that while all of the methods do a reasonable job of modeling the appearance of the person, nonetheless the competing methods fail to capture the motion dynamics well.

Moreover, in some of the aforementioned figures (such as Figures 8, 9, 10, 17, 6, 20, 26, 27 diverse sample generations by NUQ is also shown.

References

[1] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International Conference on Machine Learning*, pages 1174–1183, 2018. 2

[2] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. 2014. 3

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

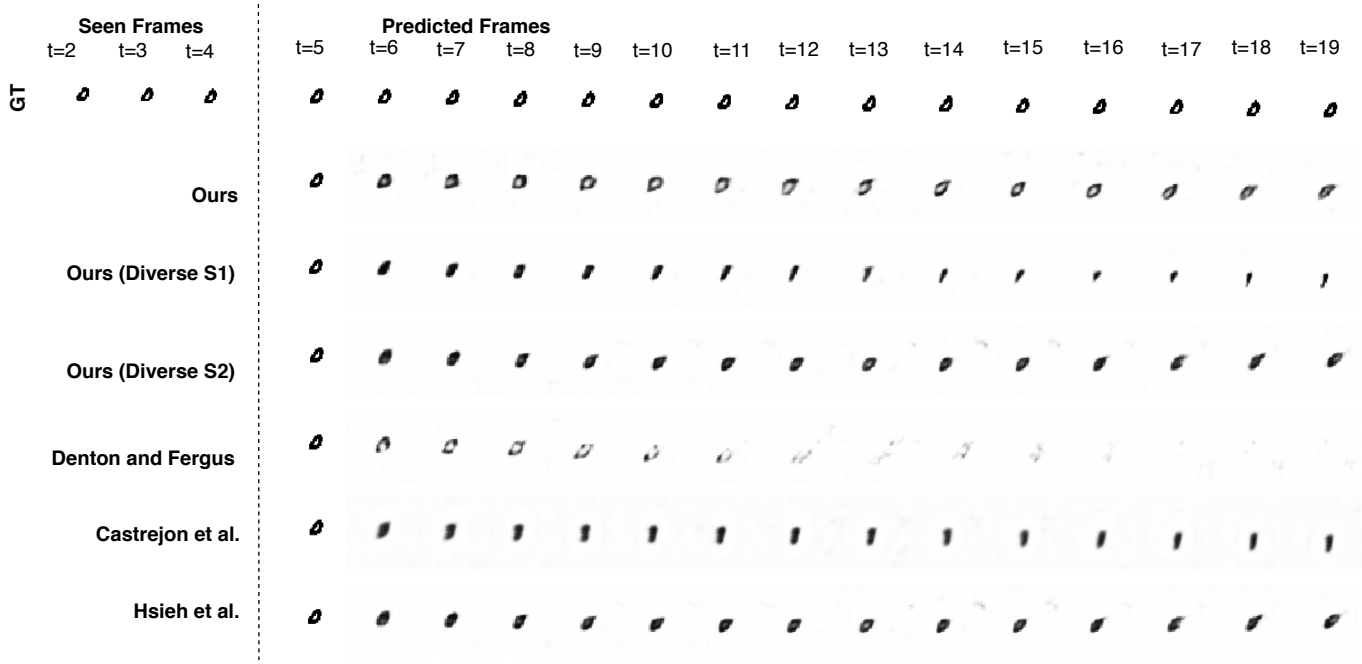


Figure 8. Visualization of generations by our method versus competing baselines on the SMMNIST Dataset, trained with 2,000 training samples. Further, diverse generations by our method are also shown.

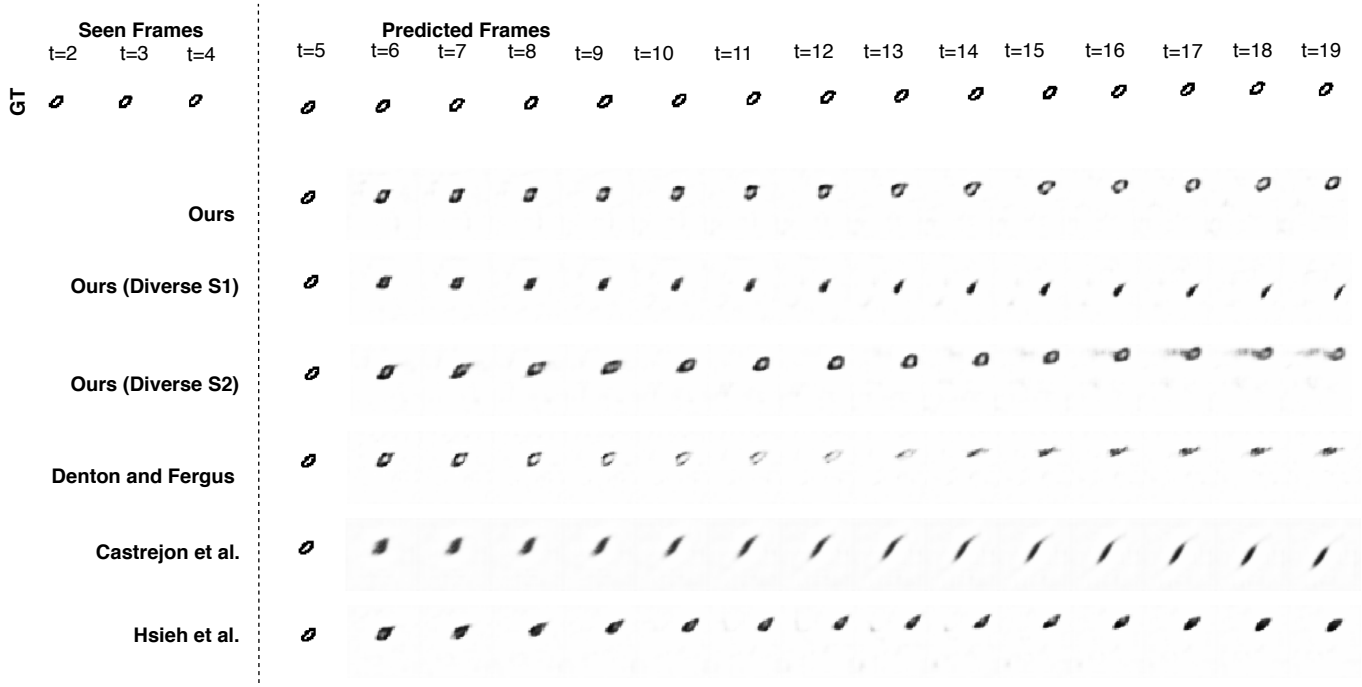


Figure 9. Visualization of generations by our method versus competing baselines on the SMMNIST Dataset, trained with 2,000 training samples. Further, diverse generations by our method are also shown.

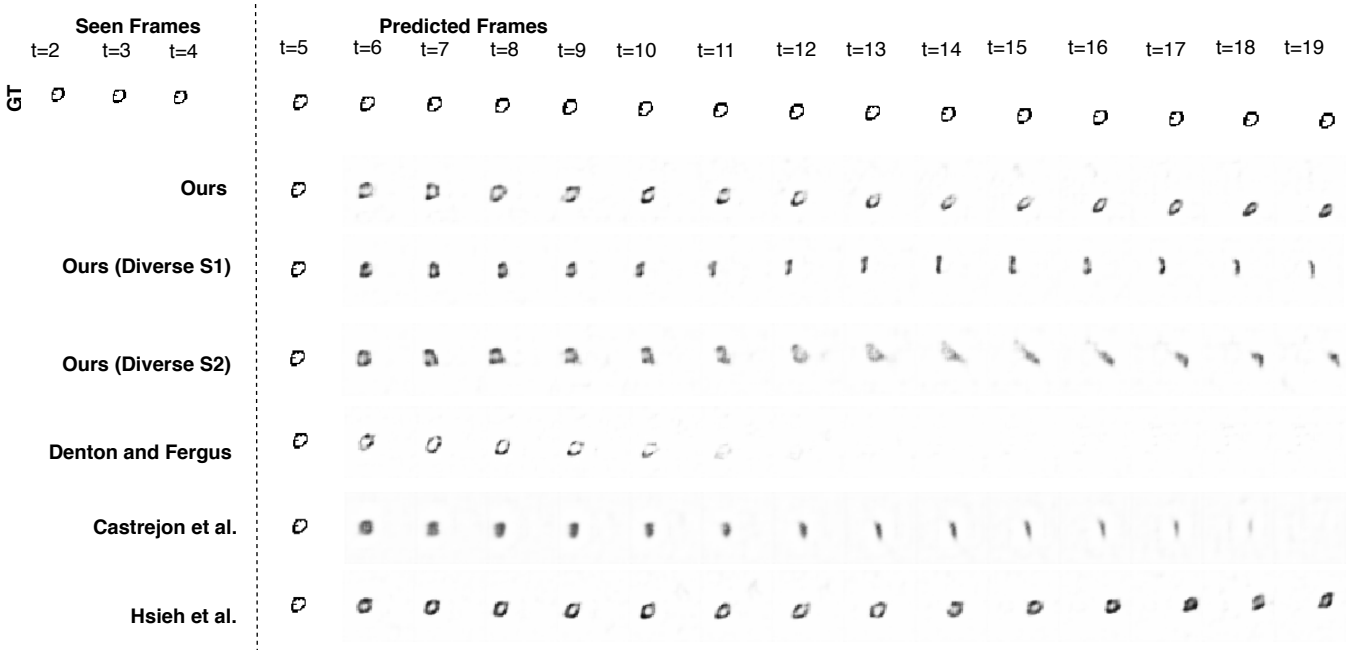


Figure 10. Visualization of generations by our method versus competing baselines on the SMMNIST Dataset, trained with 2,000 training samples. Further, diverse generations by our method are also shown.

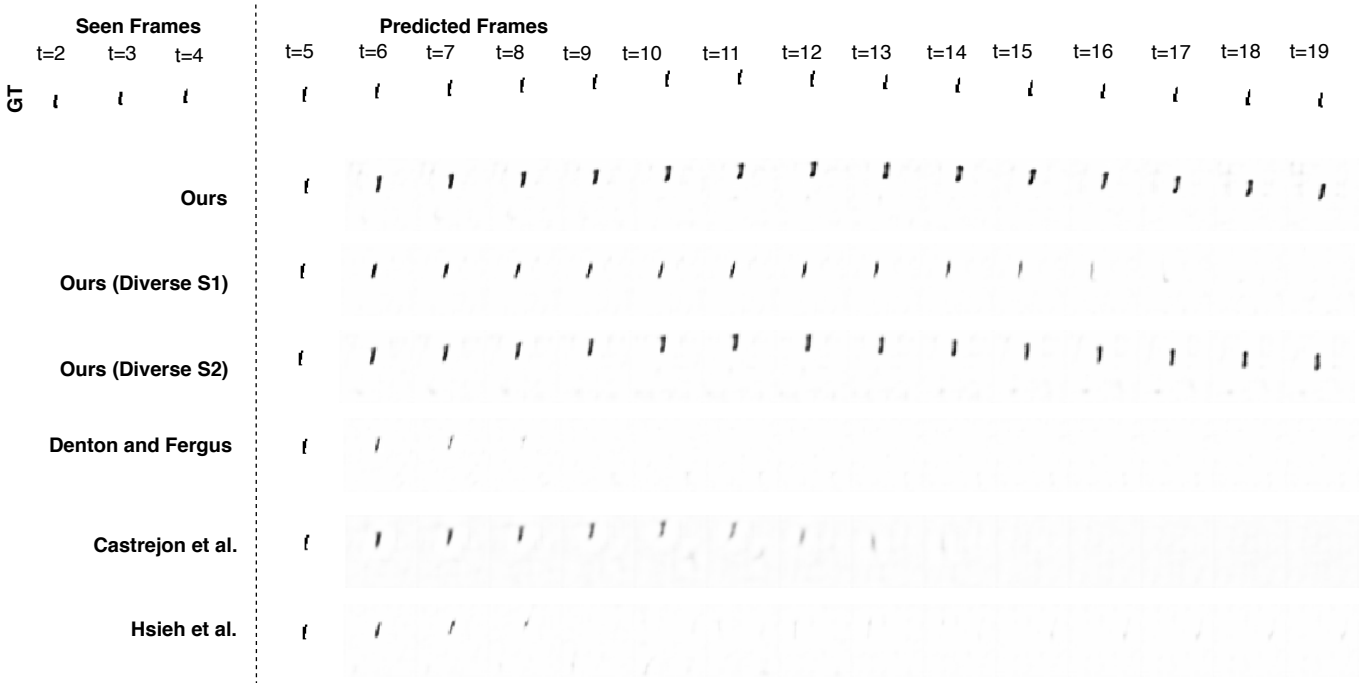


Figure 11. Visualization of generations by our method versus competing baselines on the SMMNIST Dataset, trained with 2,000 training samples. Further, diverse generations by our method are also shown.

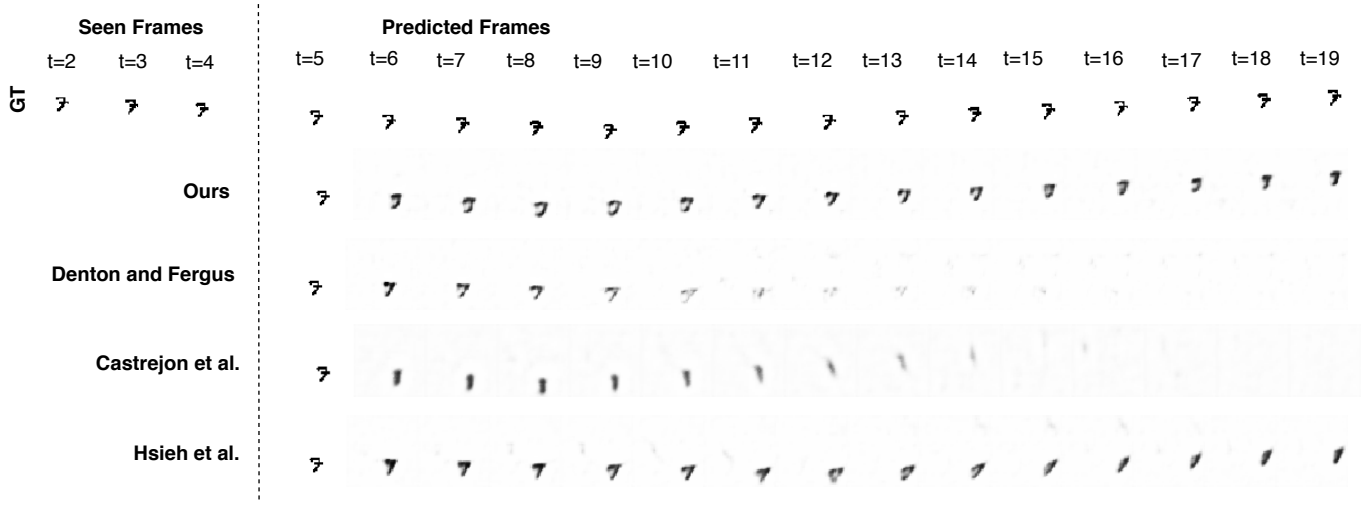


Figure 12. Visualization of generations by our method versus competing baselines on the SMMNIST Dataset, trained with 2,000 training samples.

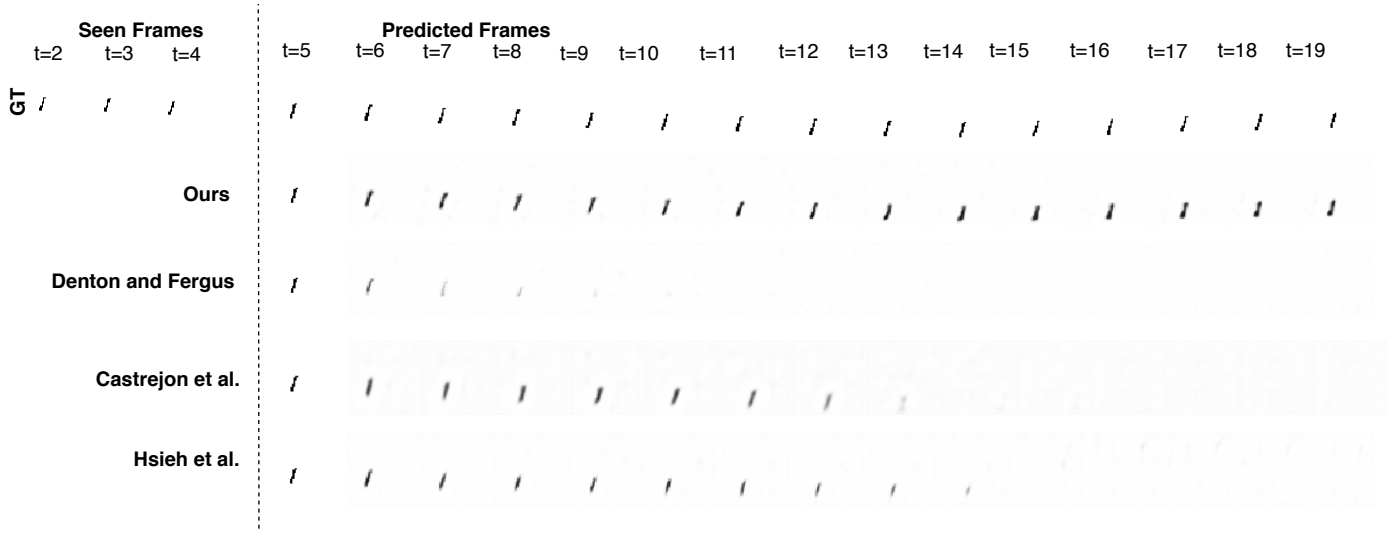


Figure 13. Visualization of generations by our method versus competing baselines on the SMMNIST Dataset, trained with 2,000 training samples.

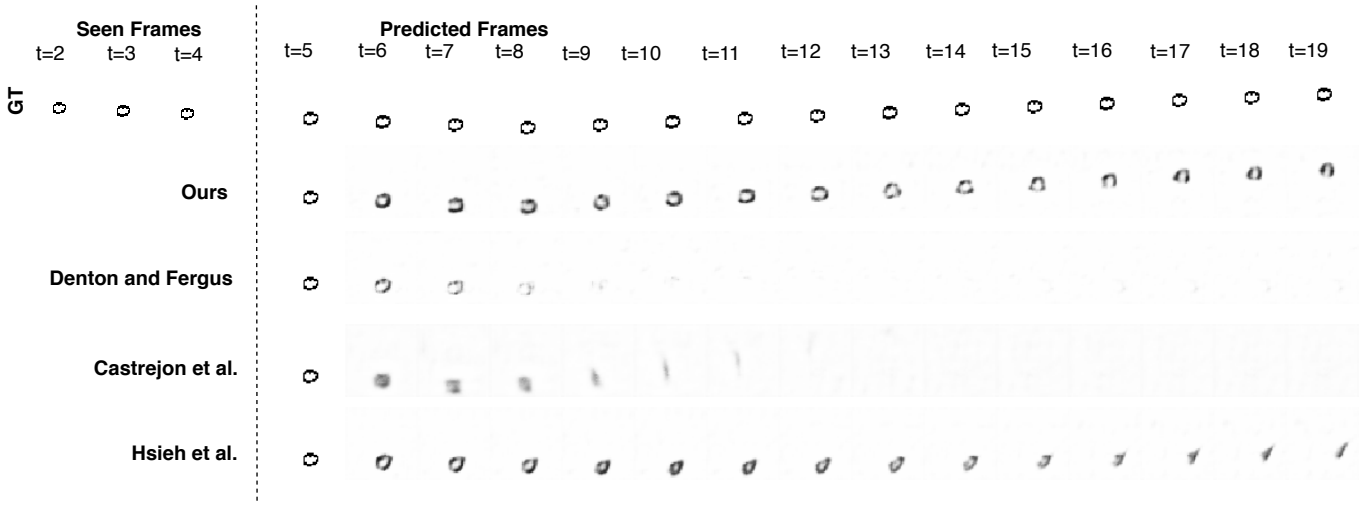


Figure 14. Visualization of generations by our method versus competing baselines on the SMMNIST Dataset, trained with 2,000 training samples.

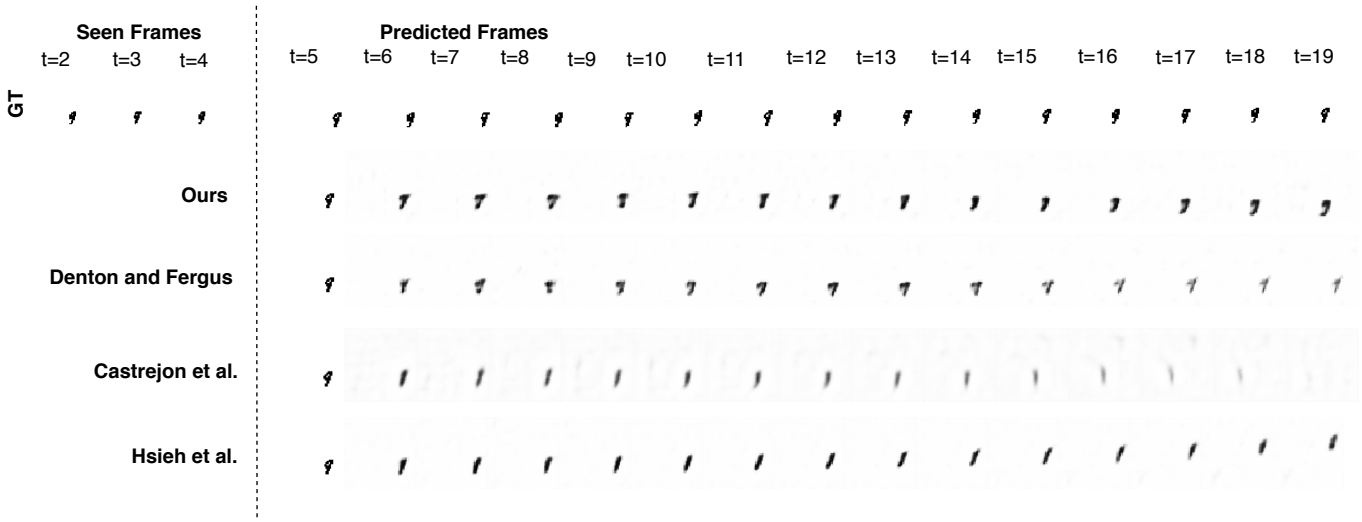


Figure 15. Visualization of generations by our method versus competing baselines on the SMMNIST Dataset, trained with 2,000 training samples.

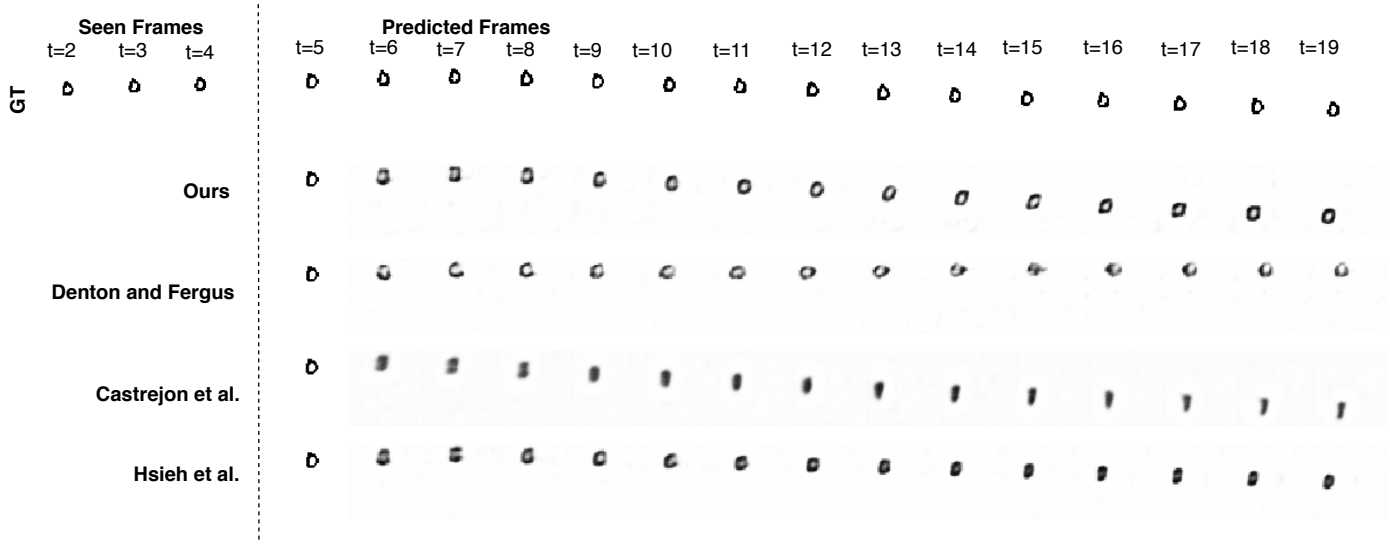


Figure 16. Visualization of generations by our method versus competing baselines on the SMMNIST Dataset, trained with 2,000 training samples.

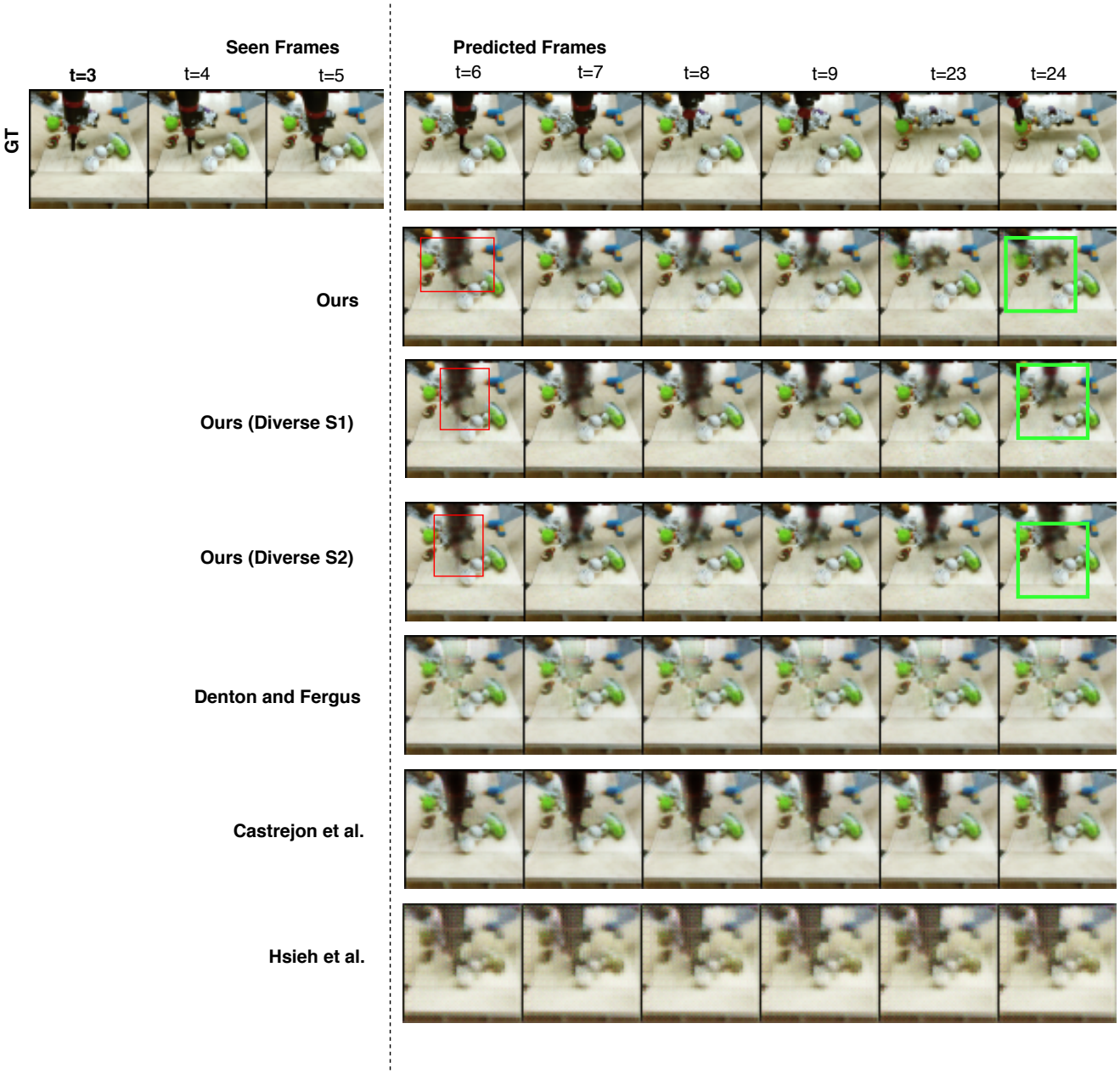


Figure 17. Visualization of generations by our method versus competing baselines on the BAIR Robot Push Dataset, trained with 2,000 training samples. Further, diverse generations by our method are also shown. High motion regions are indicated by a red bounding box, while spatial regions exhibiting high diversity are shown by a green bounding box.

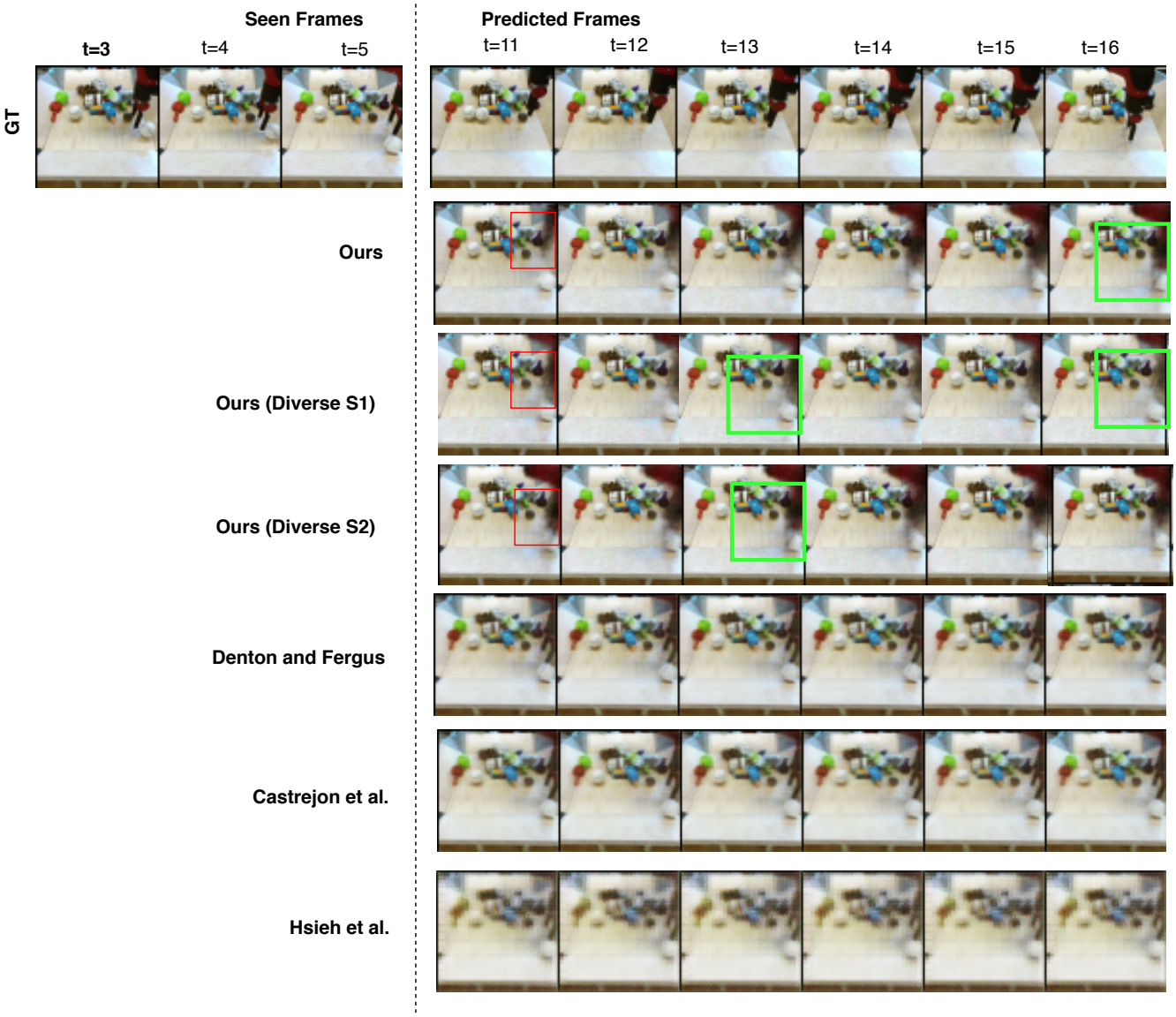


Figure 18. Visualization of generations by our method versus competing baselines on the BAIR Robot Push Dataset, trained with 2,000 training samples. Further, diverse generations by our method are also shown. High motion regions are indicated by a red bounding box, while spatial regions exhibiting high diversity are shown by a green bounding box.

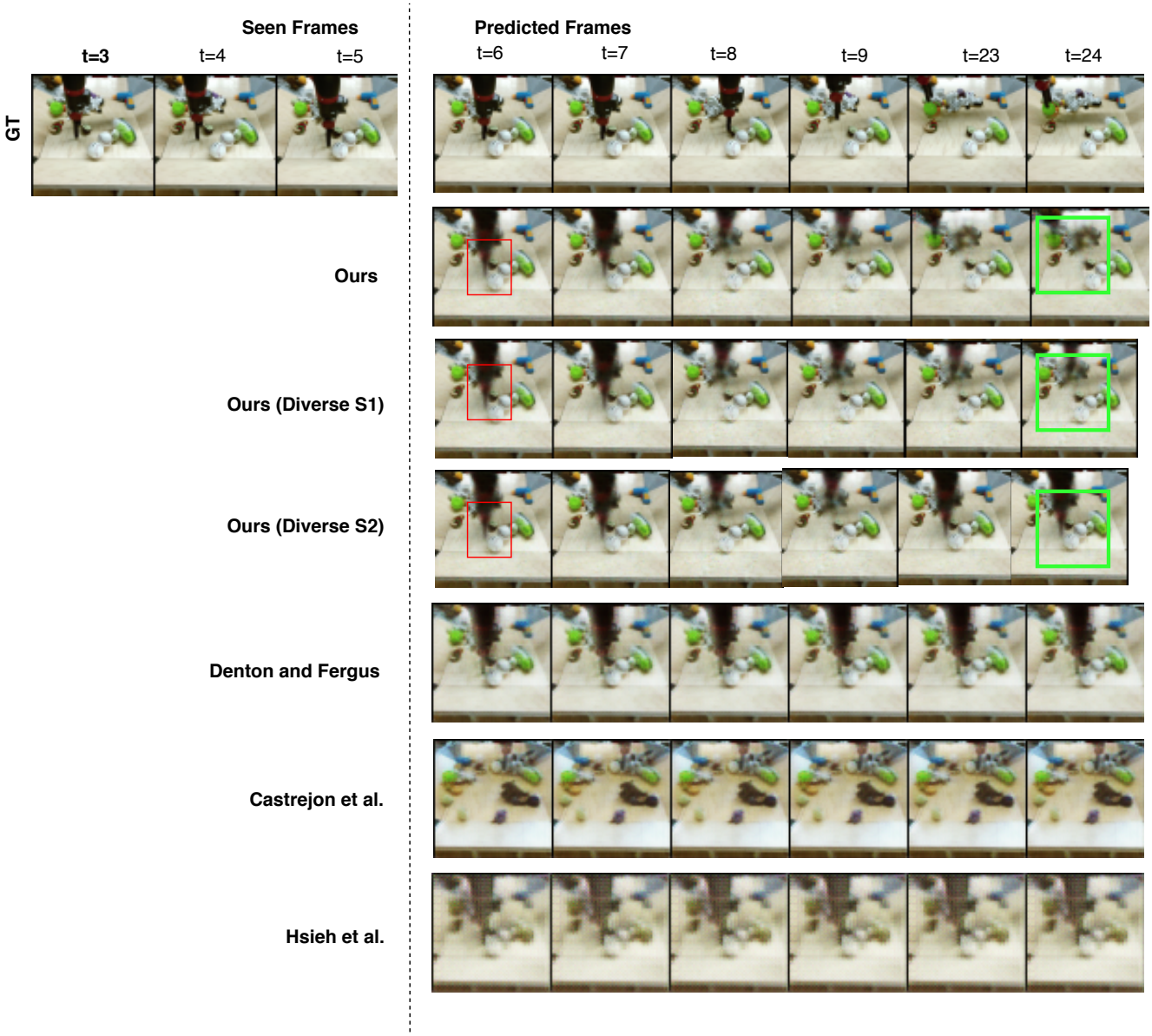


Figure 19. Visualization of generations by our method versus competing baselines on the BAIR Robot Push Dataset, trained with 2,000 training samples. Further, diverse generations by our method are also shown. High motion regions are indicated by a red bounding box, while spatial regions exhibiting high diversity are shown by a green bounding box.

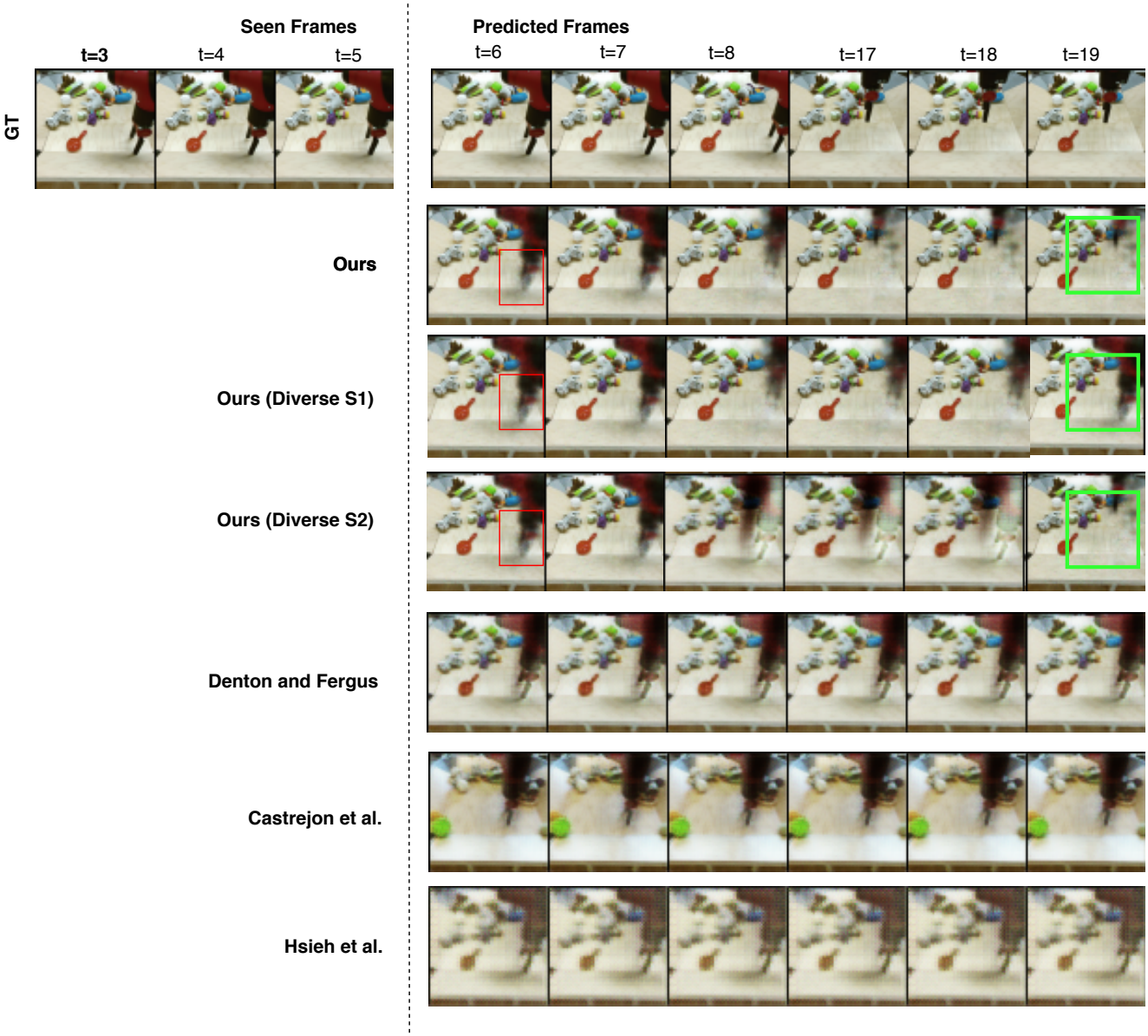


Figure 20. Visualization of generations by our method versus competing baselines on the BAIR Robot Push Dataset, trained with 2,000 training samples. Further, diverse generations by our method are also shown. High motion regions are indicated by a red bounding box, while spatial regions exhibiting high diversity are shown by a green bounding box.

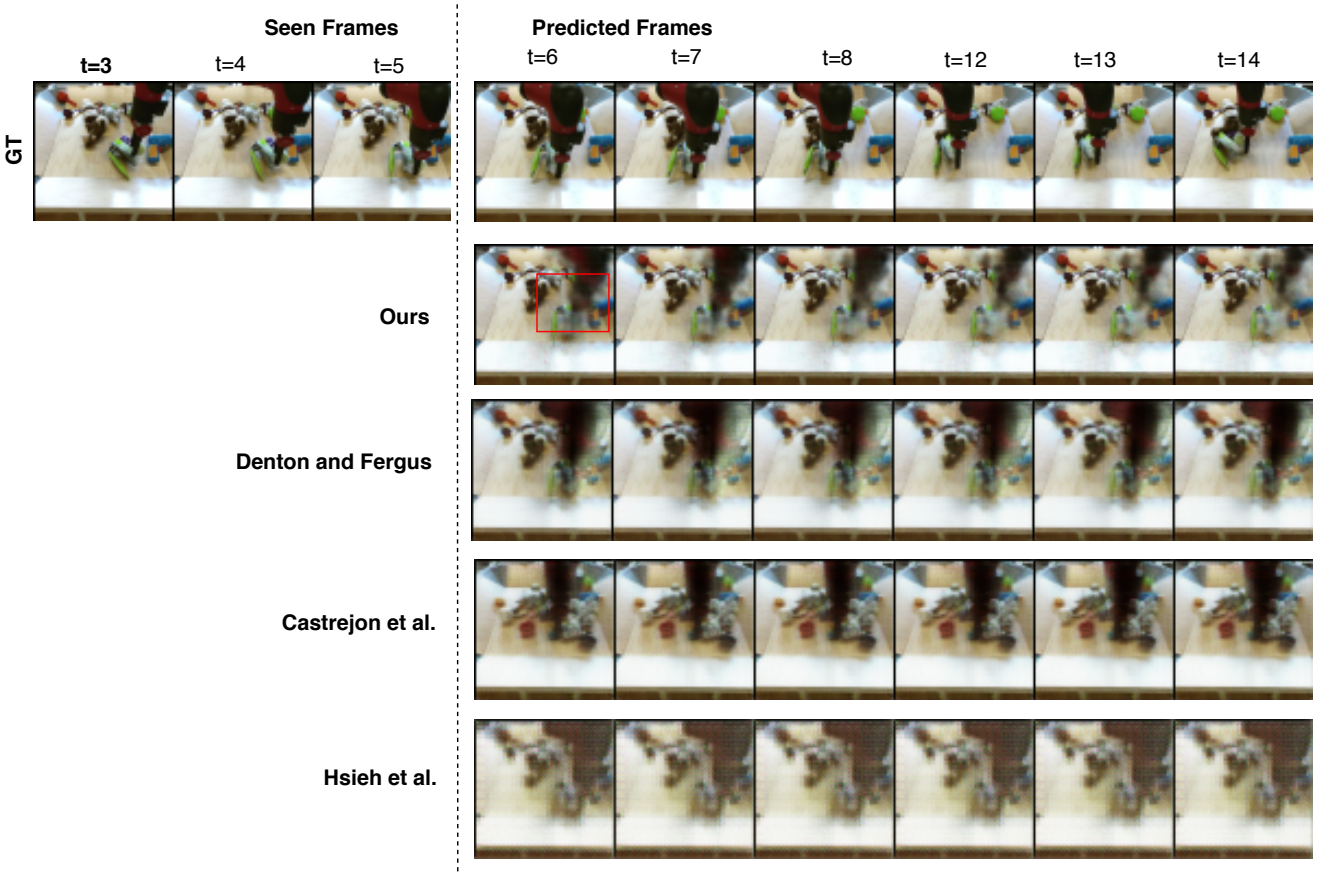


Figure 21. Visualization of generations by our method versus competing baselines on the BAIR Robot Push Dataset, trained with 2,000 training samples. High motion regions are indicated by a red bounding box.

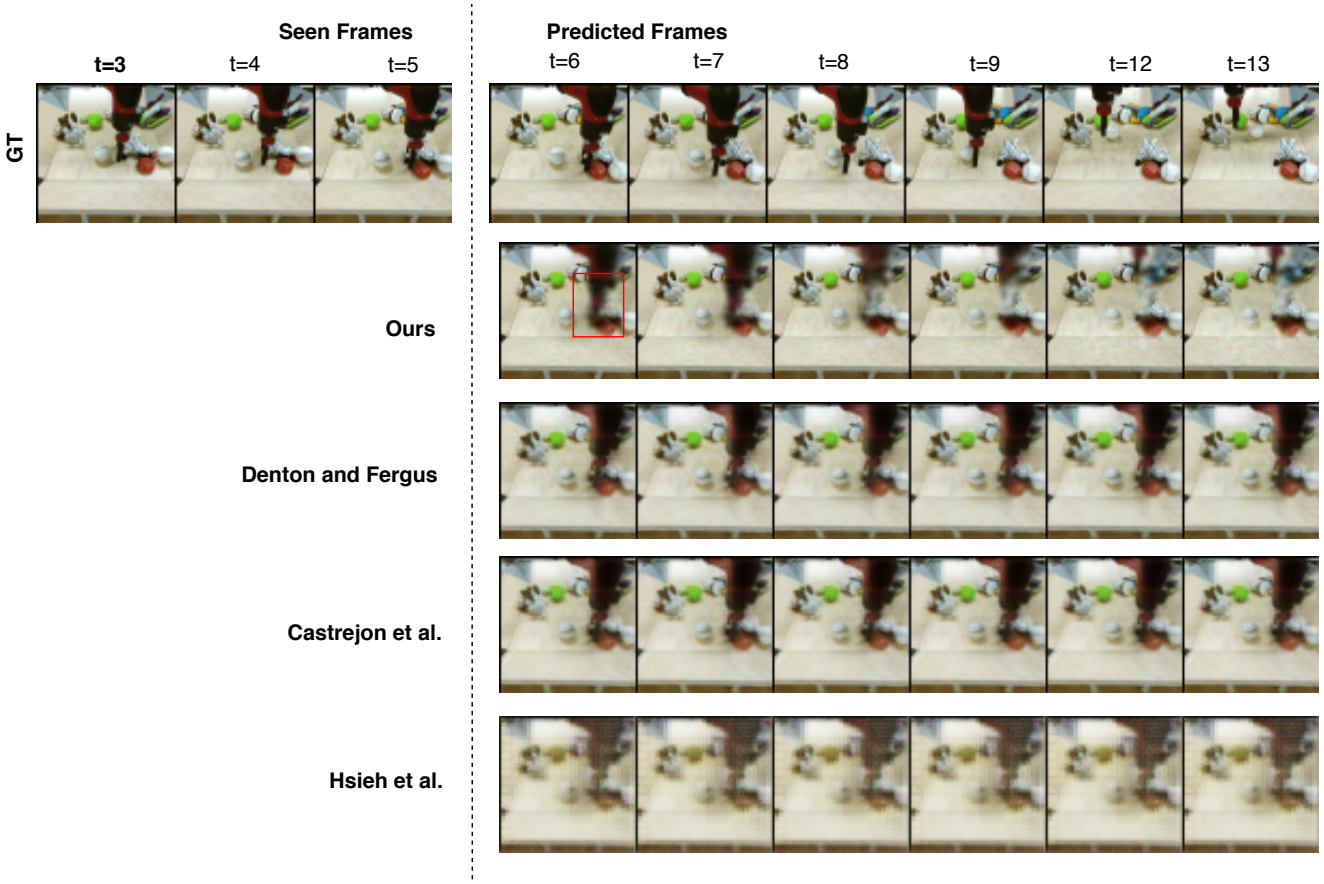


Figure 22. Visualization of generations by our method versus competing baselines on the BAIR Robot Push Dataset, trained with 2,000 training samples. High motion regions are indicated by a red bounding box.

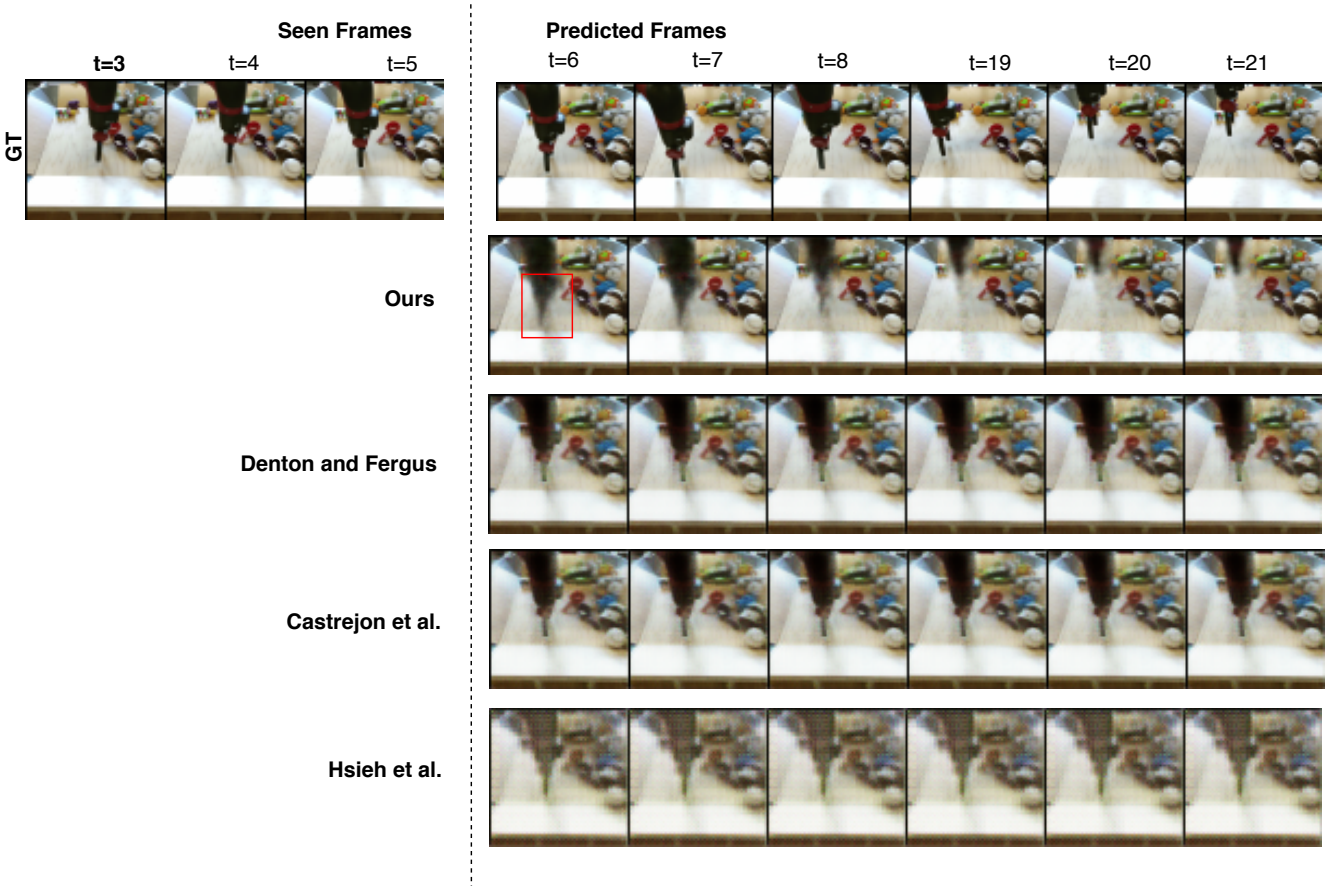


Figure 23. Visualization of generations by our method versus competing baselines on the BAIR Robot Push Dataset, trained with 2,000 training samples. High motion regions are indicated by a red bounding box.

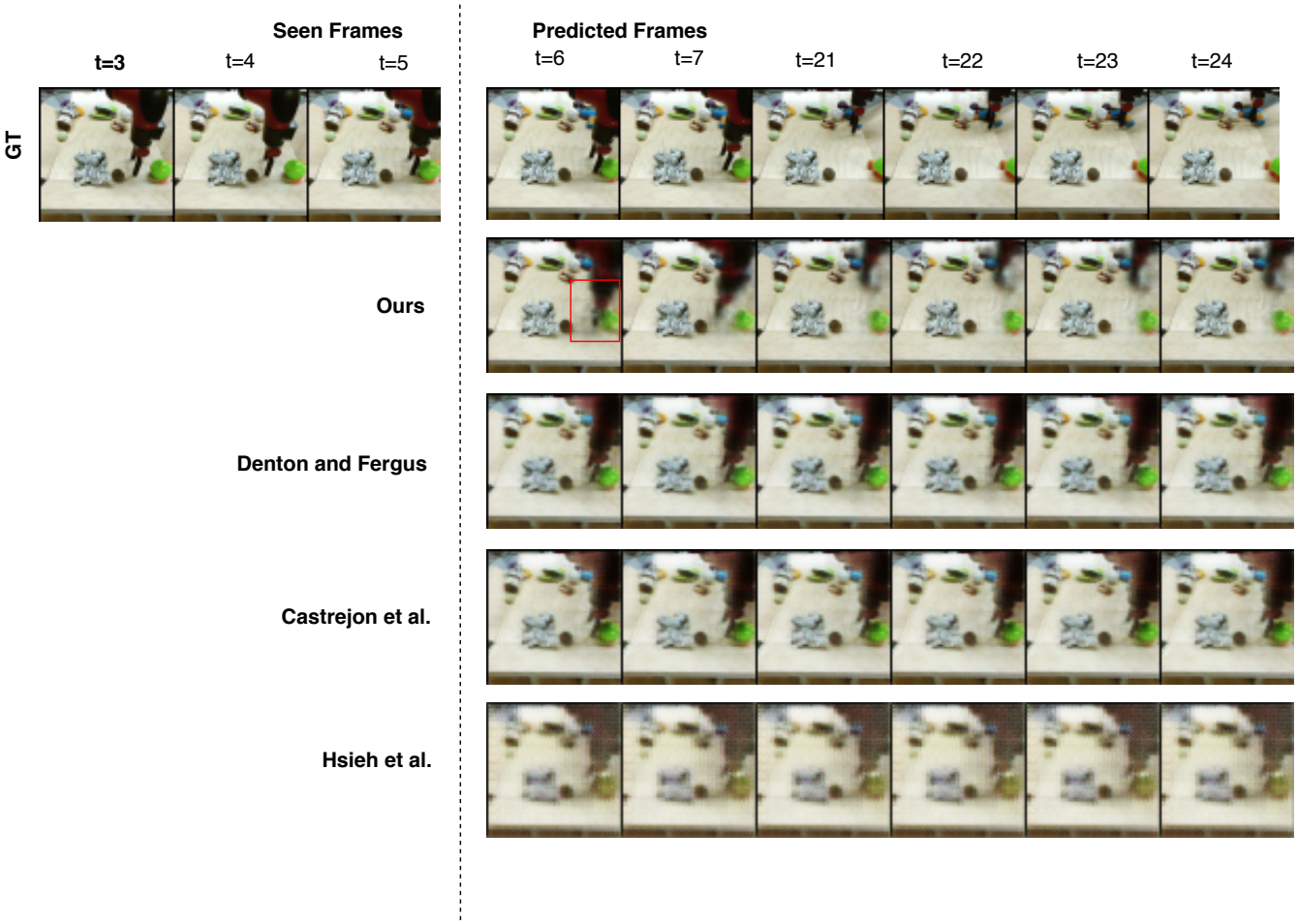


Figure 24. Visualization of generations by our method versus competing baselines on the BAIR Robot Push Dataset, trained with 2,000 training samples. High motion regions are indicated by a red bounding box.

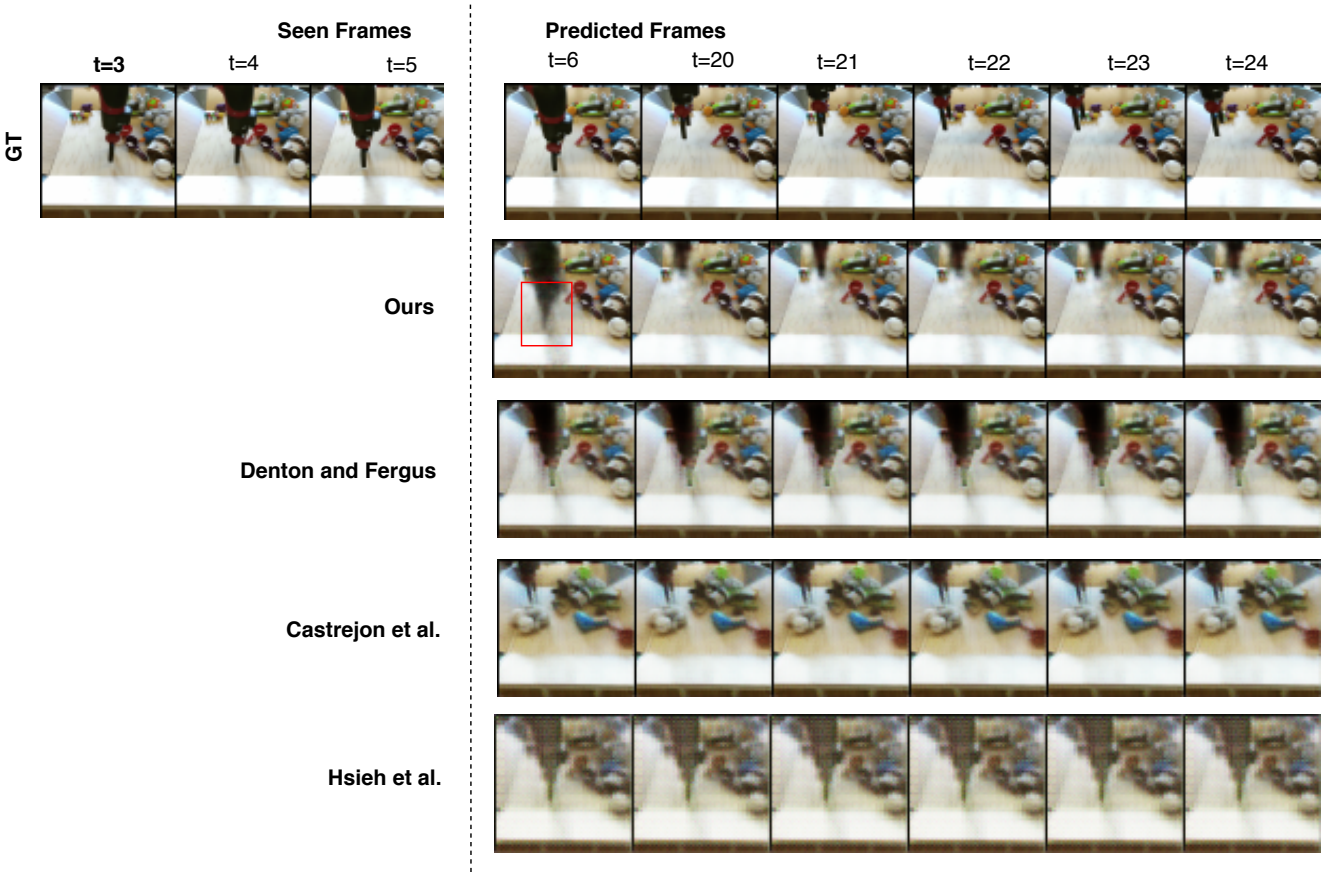


Figure 25. Visualization of generations by our method versus competing baselines on the BAIR Robot Push Dataset, trained with 2,000 training samples. High motion regions are indicated by a red bounding box.

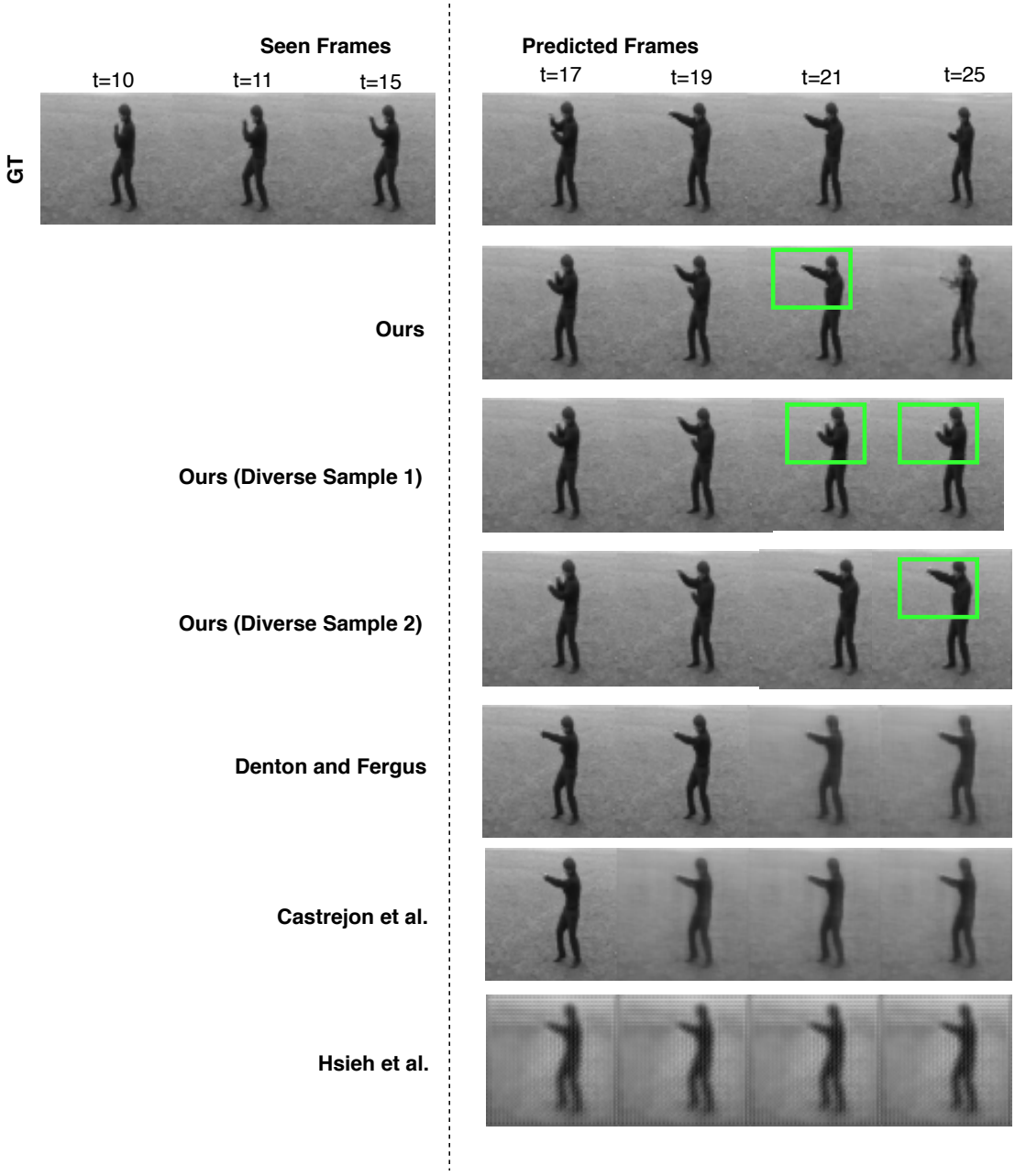


Figure 26. Visualization of generations by our method versus competing baselines on the KTH Action Dataset, trained with the full training data of 1,911 training samples. Further, diverse generations by our method are also shown. Spatial regions exhibiting high diversity are shown by a green bounding box.

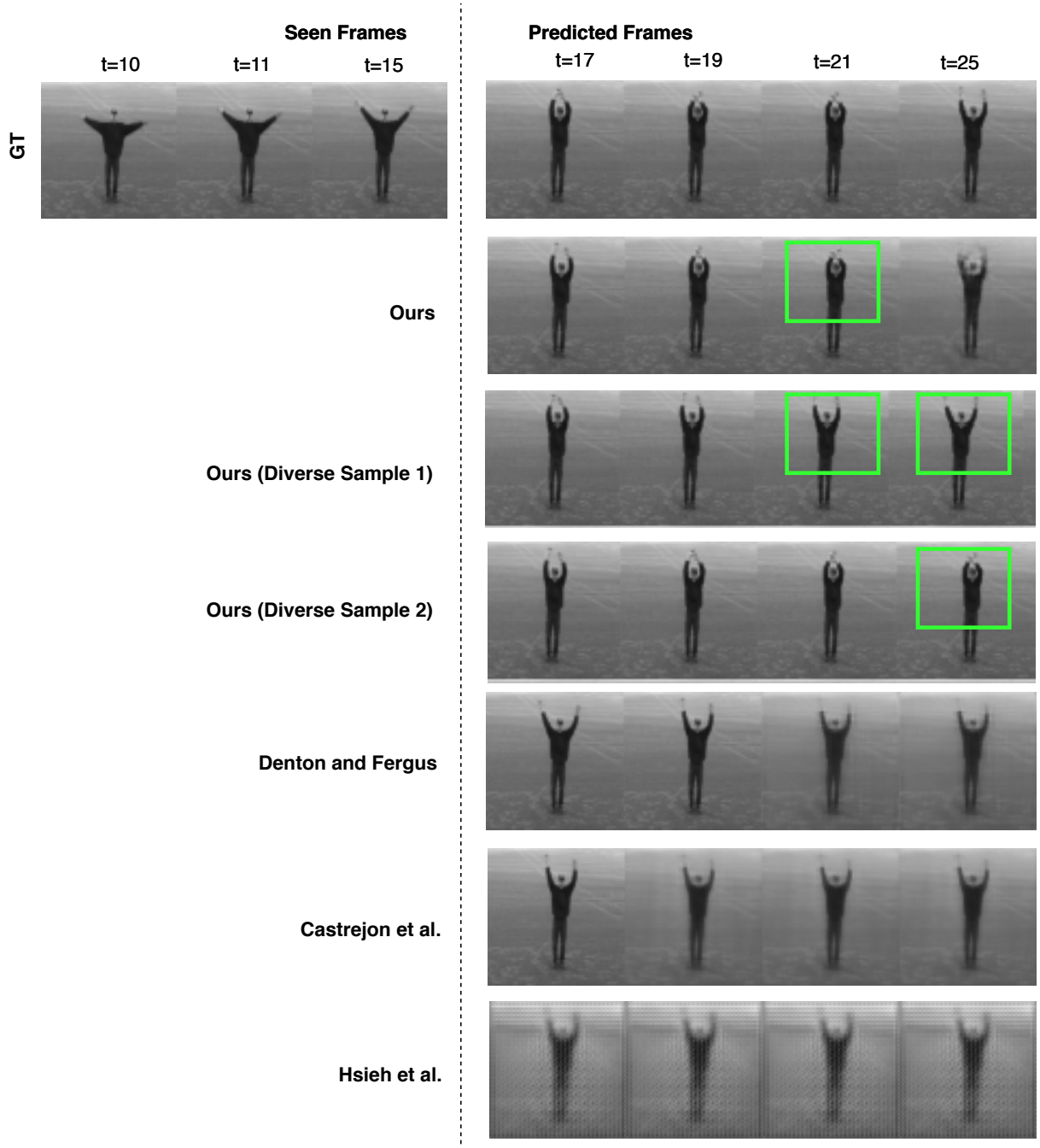


Figure 27. Visualization of generations by our method versus competing baselines on the KTH Action Dataset, trained with the full training data of 1,911 training samples. Further, diverse generations by our method are also shown. Spatial regions exhibiting high diversity are shown by a green bounding box.