

# Sparse Trajectory Modeling Experiments with SimVP\_gSTA

Claire Yang

## Abstract

The project code can be found in GitHub. This report evaluates SimVP\_gSTA on two synthetic datasets and three data representations. On geometric motion, heatmaps with WeightedMSE yield the lowest mean displacement error, while pixel-based objectives track straight lines but degrade on curves. On the Gaussian field, predictions remain near the center and do not surpass distributional guessing.

This indicates that sparse tasks either require the addition of weighted loss to achieve generalization, or apply a heatmap representation to smooth the learning. Besides, when information is insufficient, the model tends to be conservative, either outputting pure black or spreading to various possibilities while losing accuracy. Considering this, variants of Moving-MNIST is a suitable choice for future explorations with SimVP.

## 1 Objective and Summary

We evaluate whether SimVP\_gSTA can learn simple, predictable motion from very sparse inputs and identify which data representations and loss functions yield reliable forecasts. On geometric trajectories, heatmap representations with WeightedMSE achieve the lowest mean displacement error, while pixel-based WeightedBCE/MSE track straight lines but struggle on curves. On the Gaussian field, models do not improve beyond distributional guessing. We report a unified training recipe and metrics, and defer dataset limitations and implications to a dedicated section.

## 2 Dataset and Representation

We use two synthetic datasets with  $32 \times 32$  frames and one active target per frame. Each sequence has  $T = 20$  steps. All datasets are generated with random seed 42 and split into 2000/200/200 for train/val/test.

### 2.1 Gaussian Field Dataset

This dataset tests whether models can learn position-dependent motion patterns. At each time step, the point at  $p_t$  moves by a displacement sampled from a Gaussian distribution  $\Delta \sim N(s\mu, s^2\Sigma)$ , whose mean vector  $\mu$  points inward to the image center.  $s$  is a scalar constant. The covariance ellipse  $\Sigma$  is stretched along the inward axis, producing narrow stochastic trajectories. Both  $\mu$  and  $\Sigma$  depend only on position  $p_t$ .

All sequences start from  $(0, 0)$  and are clamped at the image border.

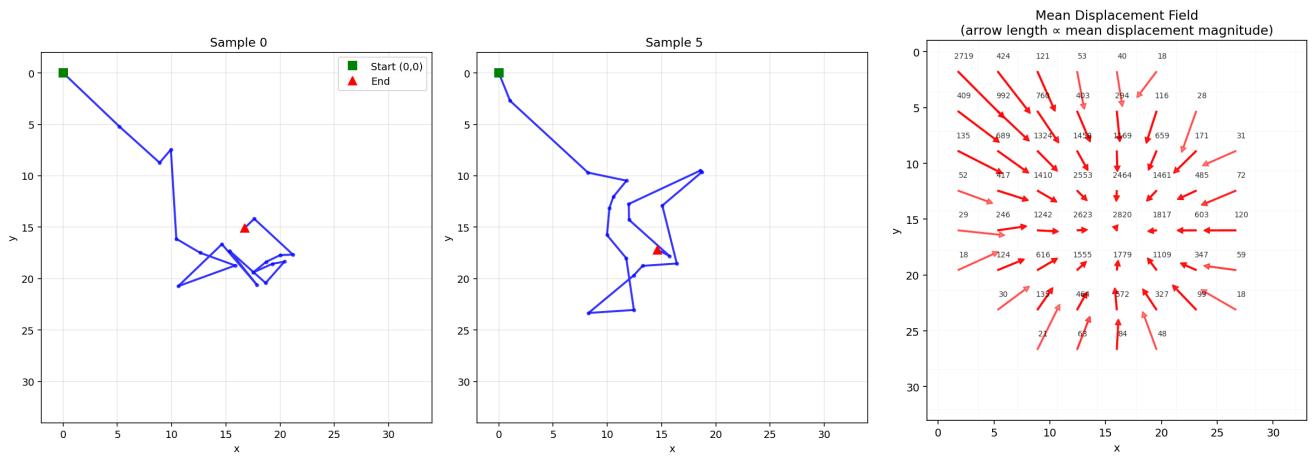


Figure 1: Left: example Gaussian trajectory (20 steps). Right: Gaussian displacement field, with a significant centripetal tendency. Arrows show average step direction and length at each grid cell. The numbers on the pixel show how many times the coordinate was sampled.

## 2.2 Geometric Pattern Dataset

This dataset contains simple deterministic motion with two types of patterns.

Line(50%): constant-velocity motion with reflection at boundaries.

Speeds:  $[0.5, 1.0, 1.5]$  px/step.

Directions:  $[0, 30, 60, 135, 150, 225, 300, 315]$  deg.

Origins: 8 positions on a  $4 \times 4$  grid.

Arc(50%): circular motion with fixed radius and angular velocity.

Radii:  $[5, 7, 10]$  px.

Angular velocities:  $[0.3, 0.6, 1.0] \times 0.1\pi$  rad/step.

Both clockwise and counter-clockwise are included.

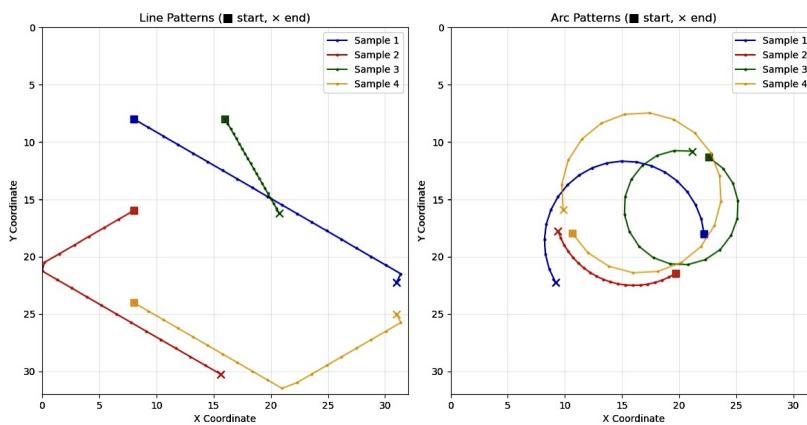


Figure 2: Examples of line and arc trajectories. Squares mark starts; crosses mark ends.

## 2.3 Data Representations

For each trajectory, there are 3 available representations.

- **Pixel:** binary  $(T, 1, 32, 32)$  with one-hot active pixel.
- **Heat:** Gaussian blur with  $\sigma = 2.0$ , rendered on  $(32, 32)$  and normalized.
- **Coord:** displacement vectors  $(T, 2)$ , stored as absolute positions.

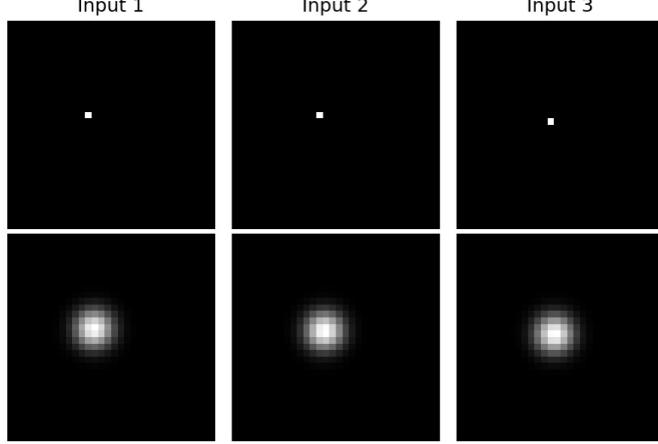


Figure 3: Top: sparse pixel frames. Bottom: Gaussian heatmaps.

### 3 Experiment Setup

We use SimVP-gSTA (OpenSTL) with 10 input and 10 prediction frames. All experiments share one training recipe:

#### Training recipe.

- Hidden dim: 64; Batch size: 32; Max epochs: 100; Seed: 42.
- Optimizer: Adam,  $lr = 10^{-3}$ , weight decay =  $10^{-4}$ .
- Early stopping: patience = 10 based on validation loss.

#### 3.1 Loss Functions

We evaluate five objectives for Pixel and Heat: WeightedBCE ( $w=1000$  for Pixel,  $w=100$  for Heat), WeightedMSE (same weights), FocalBCE ( $\alpha=1.0$ ,  $\gamma=4.0$ ), DiceBCE ( $\epsilon=10^{-6}$ ), and MSE. Full definitions are provided in the Appendix. Heatmaps use Gaussian blur with  $\sigma=2.0$ . Empirical observations: WeightedBCE handles extreme imbalance and tracks straight lines but is weak on curves; WeightedMSE is most stable on heatmaps with the lowest MDE; FocalBCE helps hard examples yet is unstable around turning; DiceBCE is ill-suited for one-pixel targets; vanilla MSE collapses on Pixels but is reasonable on Heat.

#### 3.2 Evaluation Metrics

- **Primary metric:** Mean Displacement Error (MDE@ $t$ ) for  $t=3, 6, 10$ .
- **Validity checks:** all-black ratio; freeze pattern.
- **Coordinate extraction:** Pixel via argmax; Heat via center-of-mass; Coord via cumulative sum.
- **Subset breakdown:** for Geometric dataset, we report Line and Arc separately.

## 4 Results

### 4.1 Geometric Results

**Interpretation of Table 1.** MDE@ $t$  denotes the average Euclidean displacement error over the first  $t$  prediction frames (e.g. MDE@3 averages frames 1–3), in the unit of pixel. Line MDE and Arc MDE are computed separately by splitting the test set into line and arc trajectories.

The **MSE baseline** corresponds to the original SimVP regression objective. As expected it collapses to all-black predictions and produces invalid outputs.

Among **Pixel losses**, WeightedMSE and WeightedBCE gives the best overall accuracy. FocalBCE and DiceBCE can partially track straight lines, but they fail once a trajectory changes direction. In all Pixel settings the error grows linearly with horizon length (MDE increases approximately frame by frame). The consistent gap between Line MDE and Arc MDE shows that Pixel representations struggle to capture curved motion patterns.

For **Heatmap losses**, both WeightedMSE and WeightedBCE learn stably when trajectory changes direction. The original MSE loss also achieves a low MDE of  $< 1$  pixel. They maintain low error on both line and arc subsets, with nearly identical MDE values. This indicates that the smoother spatial gradients in heatmaps allow SimVP to stabilize curved dynamics better than one-hot pixels.

Table 1: Geometric Experiments Performance Comparison

Repr	Loss	MDE@3	MDE@6	MDE@10	Line MDE	Arc MDE	Epochs
Pixel	WeightedBCE	0.951	1.119	1.426	0.903	2.003	27
	WeightedMSE	1.101	1.244	1.520	0.904	2.134	11
	FocalBCE	1.114	1.659	2.634	1.556	3.713	35
	DiceBCE	2.344	3.437	5.108	2.592	7.625	81
	MSE	-	-	-	-	-	12
Heat	WeightedMSE	<b>0.395</b>	<b>0.413</b>	<b>0.436</b>	<b>0.412</b>	<b>0.460</b>	22
	WeightedBCE	0.608	0.657	0.789	0.882	0.696	14
	MSE	0.512	0.567	0.687	0.638	0.741	23

All models that learn some patterns exhibit a roughly linear increase of error. Figure 4 shows per-frame MDE over steps for Pixel WeightedBCE as an example. This confirms that once a model captures the motion pattern, errors accumulate gradually rather than catastrophically. Also when the representation itself is smooth enough, SimVP does not necessarily require weighting to stabilize training.

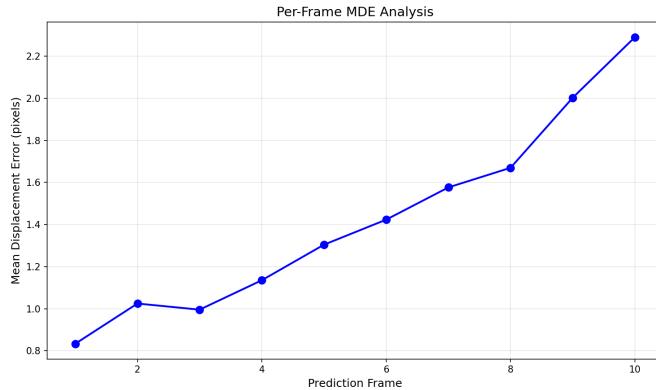


Figure 4: Per-frame MDE. Pixel models diverge quickly with horizon, while heatmaps remain more stable.

**Sample-specific analysis.** Representative frames are shown in Figure 5, with trajectories in the fourth column of Figure 6. DiceBCE and FocalBCE often collapse to all-black frames; after normalization faint signals with fairly accurate locations can be captured. WeightedBCE and WeightedMSE preserve correct positions but suffer from spreading blobs after 5–6 steps. By contrast, Heatmap WeightedMSE produces stable Gaussian spots with consistent size and intensity across frames.

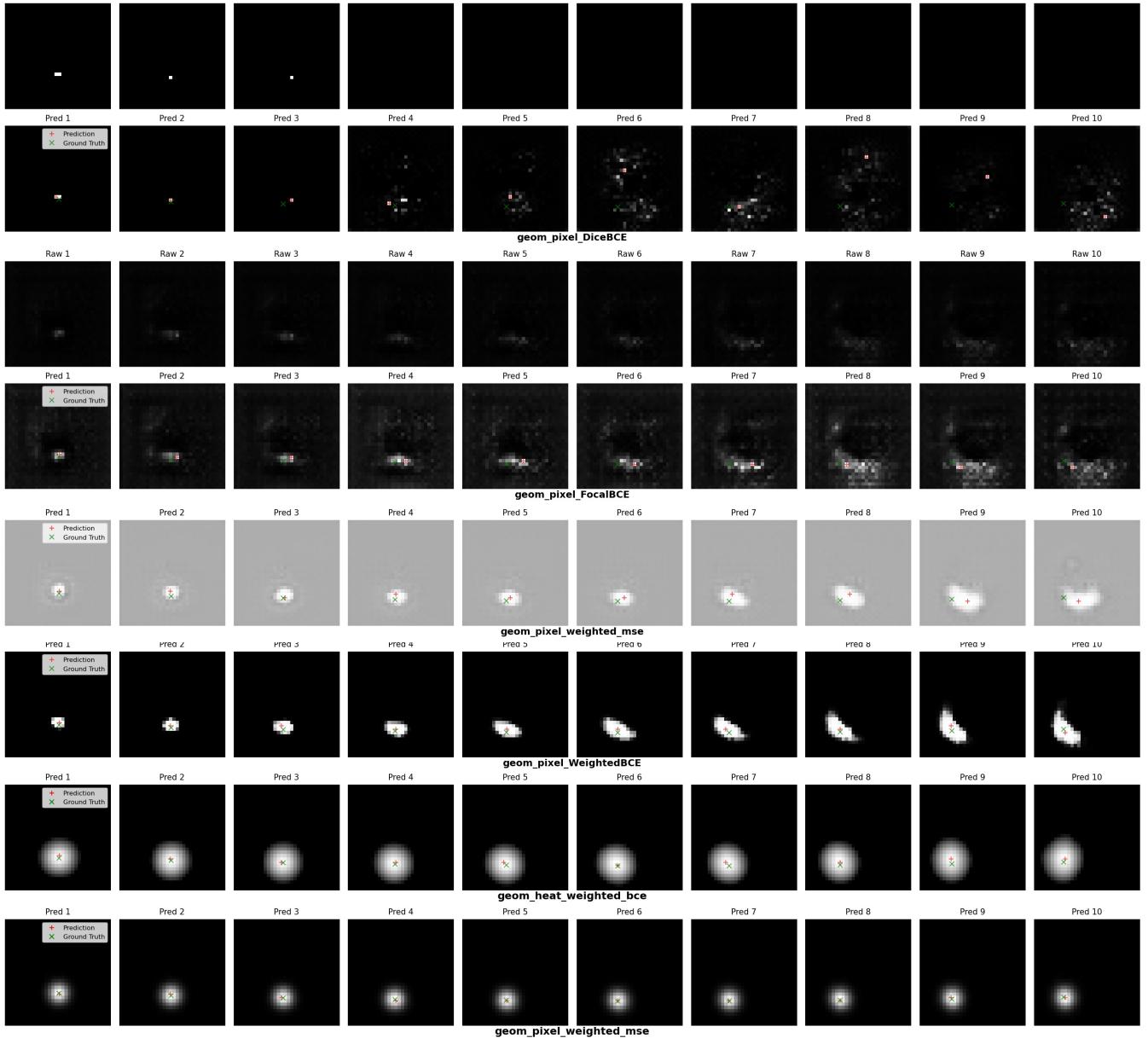


Figure 5: Frame-level predictions for Geometric dataset. Top: Pixel losses, Bottom: Heatmap losses. Normalization reveals residual predictions even when raw outputs collapse. The two rows of focalBCE and diceBCE indicating that there is a significant difference between them before and after normalization. Heatmap WeightedMSE gives sharpest and most stable spots.

Figure 6 plots predicted vs. ground-truth trajectories. Pixel models can track straight lines but fail on arcs, particularly at high angular velocity. Heatmap models closely follow both lines and arcs, indicating better pattern generalization. In the third column, where the step size is much larger, they instead perform exceptionally well, suggesting that heatmap can effectively smooth out learning. For slow straight lines, Pixel WeightedBCE captures turning points more sharply than Heatmap WeightedBCE, indicating that discrete one-hot targets can aid abrupt transitions when motion is less demanding.

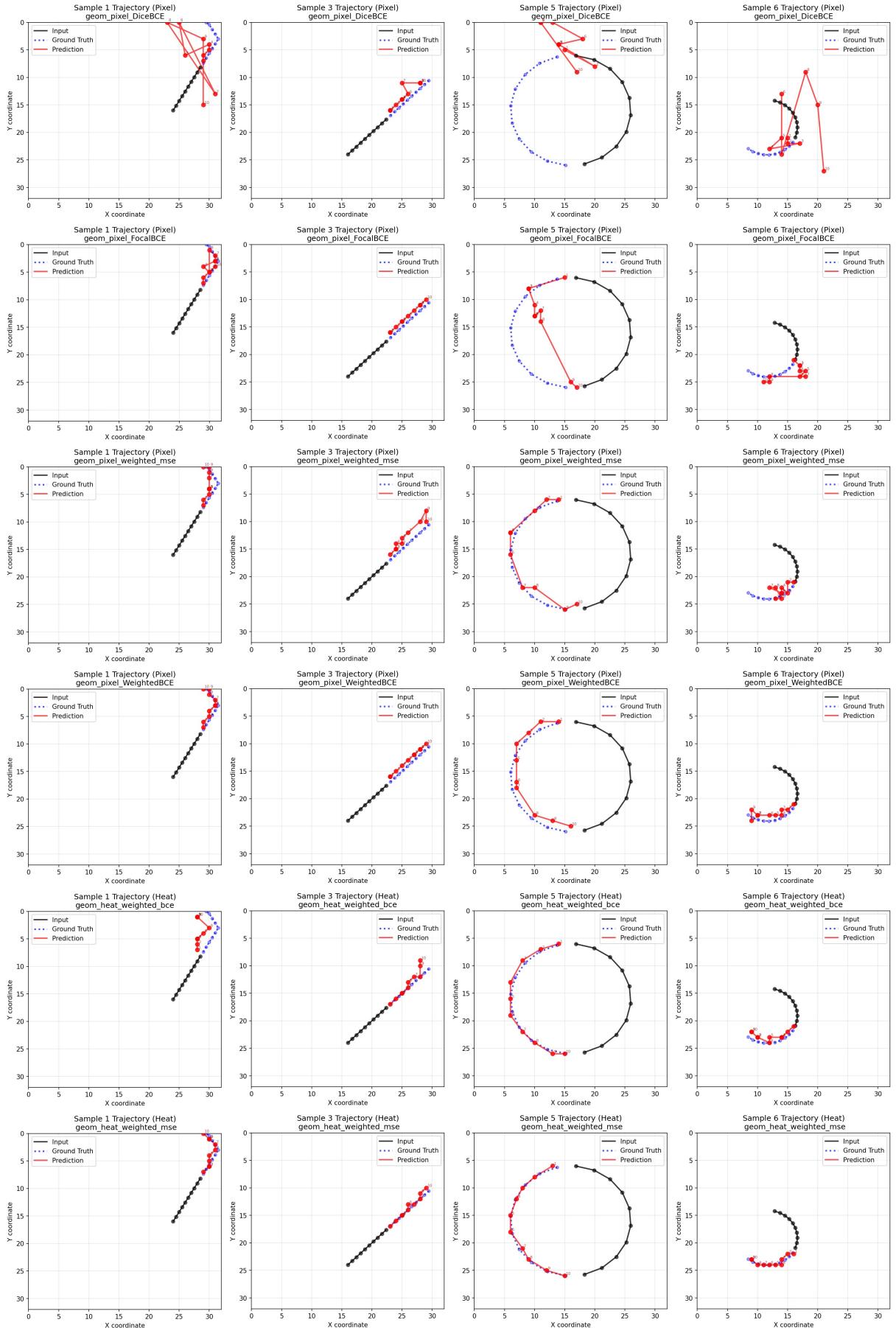


Figure 6: Predicted trajectory samples. Black = ground truth, red = prediction. Pixel models fail on arcs; heatmaps remain accurate. Pixel WeightedBCE shows strong performance on straight lines with turns.

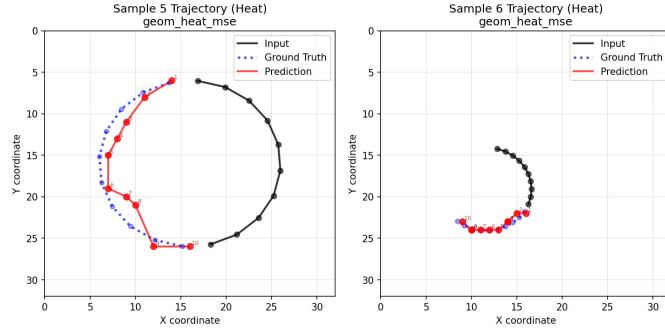


Figure 7: Additional trajectories from unweighted MSE.

Unweighted MSE handled short-step trajectories more precisely (Fig. 7), while weighted variants favored long-step arcs. This suggests that tuning the weight  $w$  or heatmap  $\sigma$  trades off short- versus long-step accuracy.

## 4.2 Gaussian Dataset Results

**Interpretation of Table 2.** On the Gaussian Field dataset, all tested models produced prediction errors around 5–6 pixels regardless of loss choice.

Table 2: Gaussian Experiments Performance Comparison

Repr	Loss	MDE@3	MDE@6	MDE@10	Epochs
Pixel	WeightedBCE	5.818	5.764	5.764	8
	WeightedMSE	5.644	5.613	5.613	8
	DiceBCE	6.433	6.017	5.904	11
Heat	WeightedMSE	5.555	5.462	5.462	14
	WeightedBCE	5.515	5.445	5.445	10

From Fig. 8, pixel-based variants often produce all-black predictions, while heatmap-based losses produce diffuse patterns that cover many plausible positions. For all representations, predictions tend not to move away from the center. Sample trajectories are shown in Fig. 9.

Taken together, these results highlight that discrete pixel encoding is less robust: when learning fails, the model defaults to all-black predictions.

**Limitations of Gaussian Data** On the Gaussian Field dataset, models quickly drift toward the center within the first few frames and then perform stochastic walks near the middle, yielding prediction errors around 5–6 pixels across objectives. The signal-to-noise ratio near the center is close to zero, leaving little learnable structure. Training commonly stops early around 10 epochs. For this reason, we change to used only the first 10 frames for training (5 input, 5 predicted), as the later frames largely reflect centered wandering.

If we need to improve, we should abandon centripetal tendency and adopt a rightward bias. However, a better option has appeared which will be discussed in next section.

## 5 Future Work

We will extend experiments to Moving MNIST. It provides dense frames and predictable piecewise-smooth motion, matching our finding that SimVP learns reliably with smooth spatial targets and handles line/arc dynamics. Unlike the Gaussian field, it avoids center-drift and weak signal issues while still testing reflections, occlusions, and multi-object interactions under controlled difficulty.

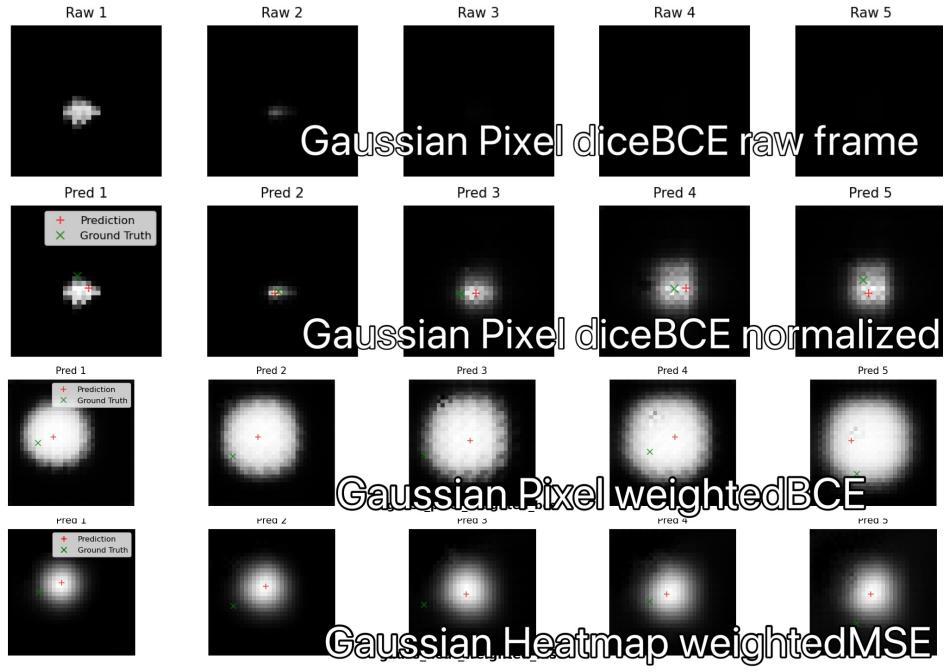


Figure 8: Frame-level predictions for Gaussian dataset. Top: Pixel losses (raw frame + normalized), Bottom: weightedBCE with pixel representation and weighted MSE with heatmap representation.

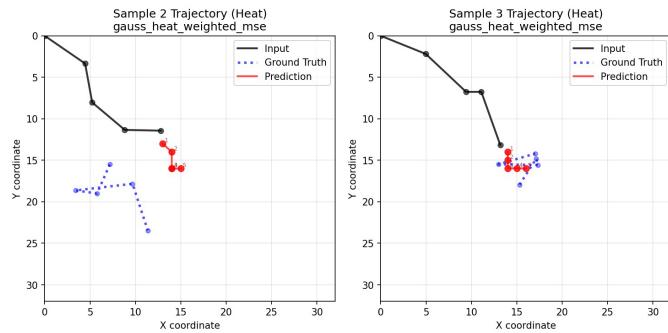


Figure 9: Two examples of trajectories of model prediction for the Gaussian dataset, trained and tested with 5 input frame and 5 output frames. These samples are generated by weightedMSE loss with heatmap representations, however all other variants in Table 2 are quite similar.

## 6 Conclusion

Heatmap representations combined with WeightedMSE train stably and achieve the lowest errors on geometric motion, maintaining similar accuracy on line and arc subsets. Pixel-based objectives can track straight lines but degrade on curves and are prone to all-black collapse when training is unstable. On the Gaussian field, none of the objectives surpass distributional guessing, which points to insufficient learnable structure in late frames. Overall, smoothing the spatial target and modest weighting are sufficient for SimVP-gSTA to learn simple dynamics; future work should evaluate dense datasets and quantify the correlation between curvature and error.

## Appendix

### Loss Definitions

Let  $y \in [0, 1]^{T \times H \times W}$  be the target,  $z$  the logits,  $p = \sigma(z)$ , and  $\langle \cdot \rangle$  denote the mean over all positions.

WeightedBCE (Pixel  $w=1000$ , Heat  $w=100$ ):  $-\langle w y \log p + (1 - y) \log(1 - p) \rangle$ .

WeightedMSE:  $\langle w_i(p - y)^2 \rangle$  with  $w_i=w$  when  $y_i>0$  else 1.

FocalBCE:  $-\langle \alpha(1 - p)^\gamma y \log p + (1 - \alpha)p^\gamma(1 - y) \log(1 - p) \rangle$ .

DiceBCE:  $1 - \frac{2 \sum p y + \epsilon}{\sum p + \sum y + \epsilon} + \text{BCE}(p, y)$ .

MSE:  $\langle (p - y)^2 \rangle$ .