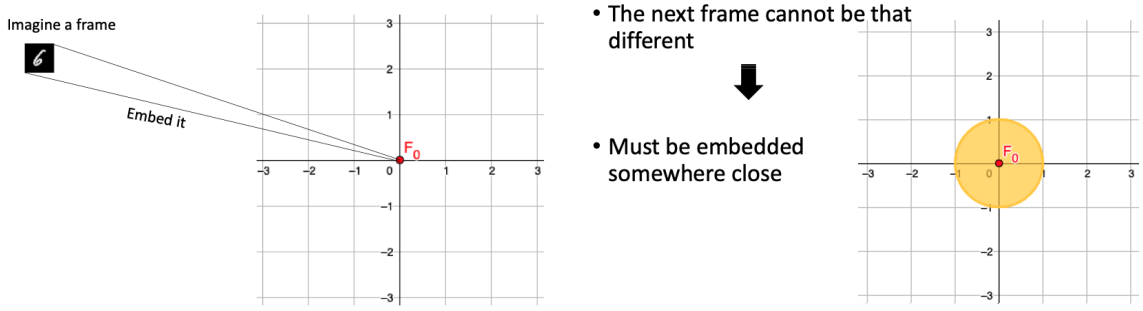
CAUSAL FUTURE PREDICTION IN A MINKOWSKI SPACE-TIME: SUPPLEMENTARY MATERIAL

Anonymous authors
Paper under double-blind review

1 STEP-BY-STEP VISUALIZATION OF THE PROPOSED ALGORITHM

In order to help the reader further their understanding of the intuition behind our proposed algorithm we will be conducting a mental experiment with the help of some visual examples. Note that for ease of understanding the figures will follow the convention of 2+1 dimensional euclidean space.

Given a frame F_0 we embed it on our space as in Figure 1a. As the next frame F_1 can only have finite differences in content compared to the first, our intuition dictates that its embedding has to lie close to the original F_0, we denote this region with yellow in Figure 1b.



(a) Lets assume a frame F_0 that we embed in our space.

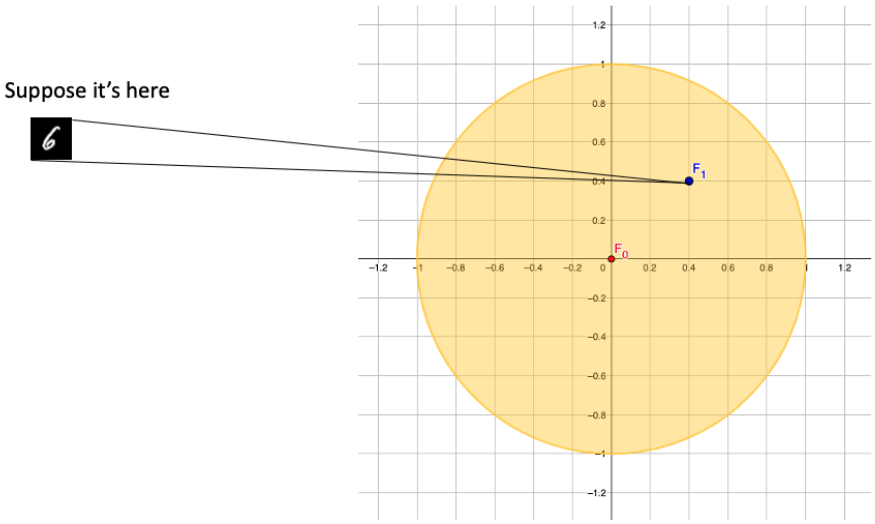
(b) The position of the next frame should be close in this space.

Figure 1

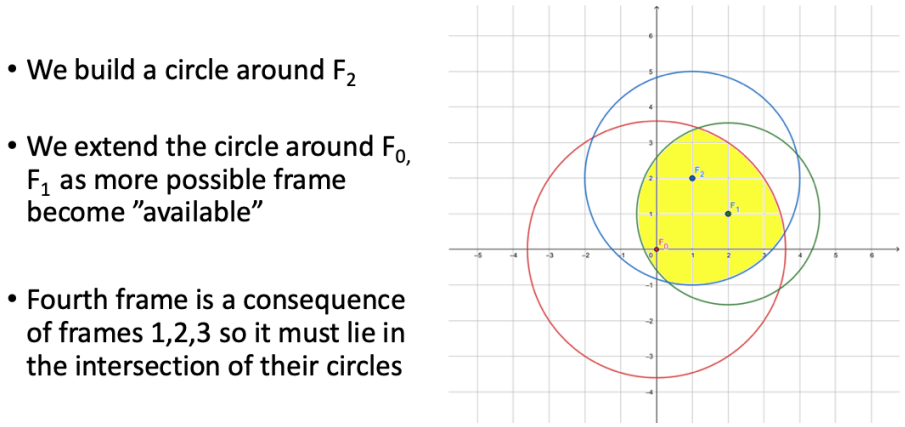
In Figure 2a we assume without loss of generality a position where F_1 will be embedded. If the next frame was known then we would simply embed it in our space in a manner similar to F_0.

Lets assume now that we have repeated the aforementioned process for a total of 3 frames and embedded them on our space. For conceptual ease, we assume frames F_0,F_1,F_2 are known a priori and we simply embed them in our space. The question that arises is where would F_3 lie? To tackle this question we have to remember that the frames constitute a sequence, hence time is also a factor that would affect our answer to the above question. In a $2D$ space we can model the passage of time by increasing the radius of the circles where the next frames lie. We base this observation on the fact that as time progresses, the content of subsequent frames can be increasingly different. Thus their embedding will be increasingly further away from our original frame F_0.

Frame F_3, however is the consequence frame of all Frames F_0,F_1,F_2, Hence given the circles of past frames are scaled accordingly to signify the passage of time from their original time t to t' in question, the new frame has to lie in the intersection of these. As seen in figure 2b



(a) Embedding of a second frame.



(b) Embedding of a third frame F_2 and enlargement of the circles of frames 0,1. A fourth frame has to lie on the intersection of the circles.

Figure 2

Our mental experiment thus far has been treating time as an invisible factor that only alters the radius of the circles. If we were to represent time as a separate observable dimension (2+1 dimensional space) then the aforementioned circles become cones, as seen in Figure 3. Hence the intersection of circles to find the constrained latent space where F_3 would lie becomes the intersection of cones. We visualize this in Figure 4

The above has been a step by step intuition of our proposed algorithm, mathematical description of the steps can be seen in Section 3 of the main paper.

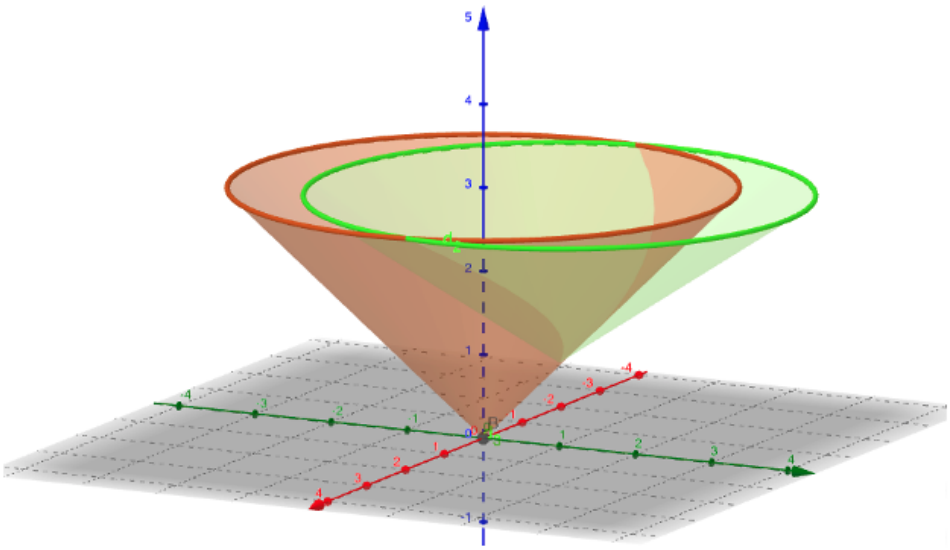


Figure 3: In space-time this would look like the intersection of cones. Due to the fact that the increasing radius of the 2D circles create a cone in 3D.

2 POINCARÉ BALL

As stated in the main paper, Riemannian Geometry can be seen as a curved generalization of Euclidean space. In this section we will be focusing on the Poincaré ball Riemannian manifold, as it forms the basis of our implementation. We note that there is no theoretical reason why we extend the Poincaré VAE Mathieu et al. (2019) other than simplicity of implementation and proven results in the image domain.

Many works rely on a Poincaré ball, as has been argued in Nickel & Kiela (2017); Mathieu et al. (2019); Ganea et al. (2018a) that embedding the latent space on a Poincaré Ball – a hyperbolic space with negative curvature – allows to naturally embed continuous hierarchical relationships between data points. This follows from the qualitative properties of such a hyperbolic space:

1. The entirety of the Poincaré Ball $\mathcal{B}_c^d$ is contained within a hypersphere of radius $1/\sqrt{c}$ and dimensionality d, in what amounts to *compactification* of infinite space.
2. The distance function (and thus area element) of this space grows rapidly as one approaches the edges of $\mathcal{B}_c^d$, such that reaching the edge would require traversing an infinite distance in latent space.
3. This behaviour naturally emulates the properties of hierarchical trees, whose size grow exponentially as new branches "grow" from previously existing branches.

Quantitatively, the space $\mathcal{B}_c^d$ is endowed with a metric tensor g^c which relates to flat Euclidean space,

$$g^c(\mathbf{r}) = \left(\frac{2}{1 - c|\mathbf{r}|^2}\right)^2 g_e(\mathbf{r}), \tag{1}$$

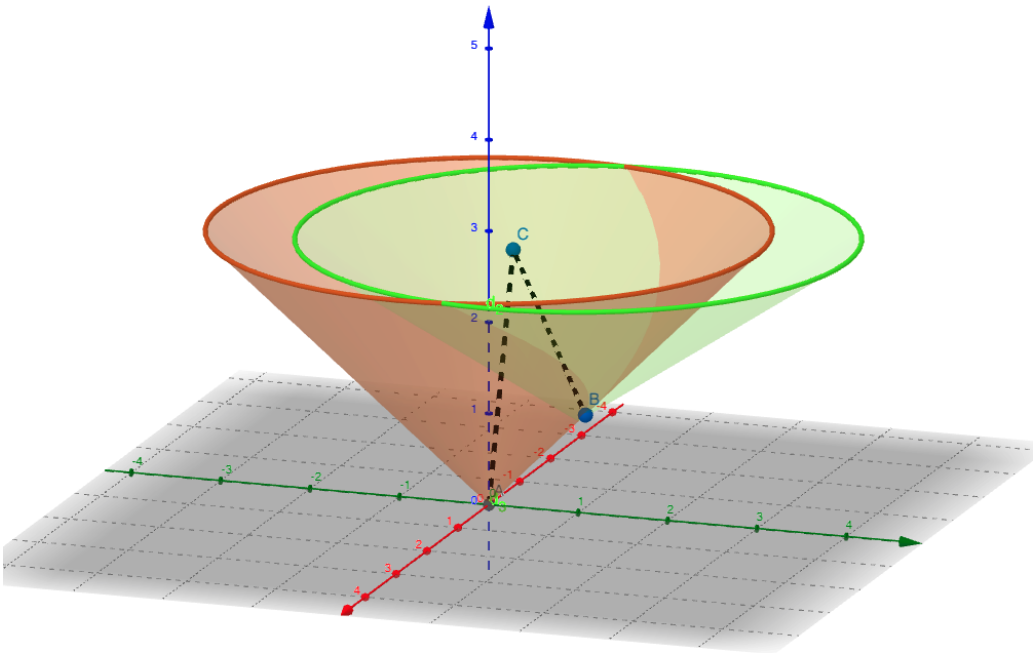


Figure 4: Two potential causal paths from points a,b to a new point c. Note that if its these points represent a sequence then the causal path will have to pass from $A \to B \to C$

where $\mathbf{r}$ is a d-dimensional vector in latent space and $g_e(\mathbf{r})$ the Euclidean metric. As a result, the distance element in $\mathcal{B}_c^d$ may be written, in spherical coordinates,

$$ds^2 = \left(\frac{2}{1-cr^2}\right)^2 (dr^2 + r^2 d\Omega_d^2), \tag{2}$$

where $r = |\mathbf{r}|$ is the radius from the origin of the space and $d\Omega_d$ is the differential solid angle element in d dimensions. It easy to see that the distance element diverges as $r \to 1/\sqrt{c}$, thus encoding the infinite hypervolume contained near the edges of the Poincaré Ball. Furthermore, as $c \to 0$, the radius of the Poincaré Ball becomes infinity and $g_c(\mathbf{z}) \to g_e(\mathbf{z})$, up to a constant rescaling of the coordinates. Let $\gamma : t \to \gamma(t)$ be a curve in $\mathcal{B}_c^d$, where $t \in [0, 1]$ such that its length is defined by

$$L(\gamma(t)) = \int_0^1 \sqrt{ds^2(t)}dt = \int_0^1 \sqrt{\mathbf{v}^T(t)\, \hat{\mathbf{g}}^{\mathbf{c}}\, \mathbf{v}(t)}\, dt, \tag{3}$$

where $\hat{\mathbf{g}}^{\mathbf{c}}$ is the matrix form of g^c and $\mathbf{v} \equiv \frac{d\mathbf{r}}{dt}$ is the trajectory's velocity vector. In component form, this reads

$$= \int_0^1 \sqrt{\sum_{\mu=1}^{d}\sum_{\nu=1}^{d} \frac{dx^\mu(t)}{dt} g^c_{\mu\nu} \frac{dx^\nu(t)}{dt}}\, dt = \int_0^1 \left(\frac{2}{1-cr^2(t)}\right)^2 \sqrt{\mathbf{v}^T\mathbf{v}}\, dt, \tag{4}$$

where x^μ represents each coordinate, $g^c_{\mu\nu}$ is the component form of $\hat{\mathbf{g}}^{\mathbf{c}}$ and in the last step we have used in Eq. (1). In hyperbolic space, "straight lines" are defined by *geodesics* $\gamma_g(t)$, *i.e.*, curves of constant speed

and least distance between points $\mathbf{x}$ and $\mathbf{y}$. Thus,

$$\gamma_g(t) = \text{argmin}\left[L(\gamma(t))\right]_{\gamma(0)=\mathbf{x}}^{\gamma(1)=\mathbf{y}} \quad \text{and} \quad \left|\frac{d\gamma(t)}{dt}\right| = 1. \tag{5}$$

With Eq. (4) and Eq. (5), one may show that the distance function $d^c(\mathbf{x}, \mathbf{y})$ between two points $\mathbf{x}$ and $\mathbf{y}$ on $\mathcal{B}_c^d$ can be computed to yield

$$d^c(\mathbf{x}, \mathbf{y}) = \frac{1}{\sqrt{c}}\text{arccosh}\left(1 + 2c\frac{|\mathbf{x}-\mathbf{y}|^2}{(1-c|\mathbf{x}|^2)(1-c|\mathbf{y}|^2)}\right). \tag{6}$$

3 WRAPPED NORMAL

While embedding data on a Riemannian space with the use of a Riemannian VAE, it is important to embed the used distribution in this space as well. Multiple ways have been proposed to perform this operation. We follow the wrapped normal distribution approach Grattarola et al. (2019); Mathieu et al. (2019).

For this, a normal distribution is mapped onto the manifold using the manifold's exponential map. Given a normal distribution $z_e \sim \mathcal{N}(0, \Sigma)$ and the Riemannian sample $z = exp_\mu^c(\frac{z_e}{\lambda_\mu^c})$, the distribution's density can be described as

$$\mathcal{N}_{\mathcal{B}_c^d}^W(z|\mu, \Sigma) = \frac{dv^W(z|\mu,\Sigma)}{d\mathcal{M}(z)} = \mathcal{N}(\lambda_\mu^c log_\mu(z)|0, \Sigma)(\frac{\sqrt{c}d_p^c(\mu, z)}{sinh(\sqrt{c}d_p^c(\mu, z))})^{d-1}. \tag{7}$$

With $c \to 0$ the Euclidean normal distribution can be obtained.

4 ARCHITECTURAL CONSIDERATIONS

In order to properly embed information on a manifold, a set of considerations have to be taken into account as developed by Ganea et al. Ganea et al. (2018b). In this paper we are following the architectural guidance of Ganea et al. (2018b); Mathieu et al. (2019) regarding the last layer of the encoder and the first layer of the decoder. Specifically in the encoder we use the Fréchet mean as calculated by the exponential mapping exp_0^c and a solftplus variance σ. In terms of the decoder we utilize the gyroplane layer as developed by Ganea et al. (2018b); Mathieu et al. (2019). Our architecture follows the consideration from Mathieu et al. (2019) with the additions of the extended mapping using equations 5,6 from the main paper and increased capacity of the hidden layers as our input is 32×32 rather than the original 28×28.

Optimization: In terms of optimization we tested both a Riemannian stochastic gradient descent as seen in Nickel & Kiela (2017) and Mathieu et al. (2019). These approaches use the exponential mapping to bring the model's parameters onto the manifold. As there were inconsequential practical differences between the two methods and as both are theoretically sound we opted for Mathieu et al. (2019)'s approach to maintain computational simplicity.

5 ONLINE LEARNING AND ANOMALY DETECTION

A limitation of current causal analysis methods is that they fail to include unseen causal sources. For example in the case of an autonomous vehicle simulation the addition of a second car by the researcher would entail a causal anomaly in the world model of the autonomous vehicle where it used to be the only vehicle in existence. It is obvious that causal future predictions are impossible to perform in such a case as the system is assumed to be impervious to modifications from outside sources.

However, by presenting an anomaly like that and mapping it on our world model we are able to raise an anomaly flag. If an observation is made that falls outside the perceived light cone of the system then if an event like this has happened before we are able to adapt the aperture of the cone, or as in the case of the scientist inserting a new vehicle, re-structure the world model of the system based on the new observation. This can be considered as a method of online fine-tuning. We believe that beyond fully retraining our model we are able to adjust the embedding space and subsequently the light cones by modifying the metric with the use of free parameters. Investigations on which method is optimal are intriguing for future work.

6 CODE

The proposed algorithm was implemented in PyTorch and the code will be made available by the time of the conference.

7 EXTRA EXPERIMENTAL RESULTS

Below we showcase a few more experimental results related to Exp. 2, 3 of the main paper. In Figure 5 we show 2 more cases of the handwaving action and 1 of the walking action. We note that the walking action depicts direction 3 of the KTH dataset - an away from the camera movement. We further show digits $7, 8$ at a variety of movements.

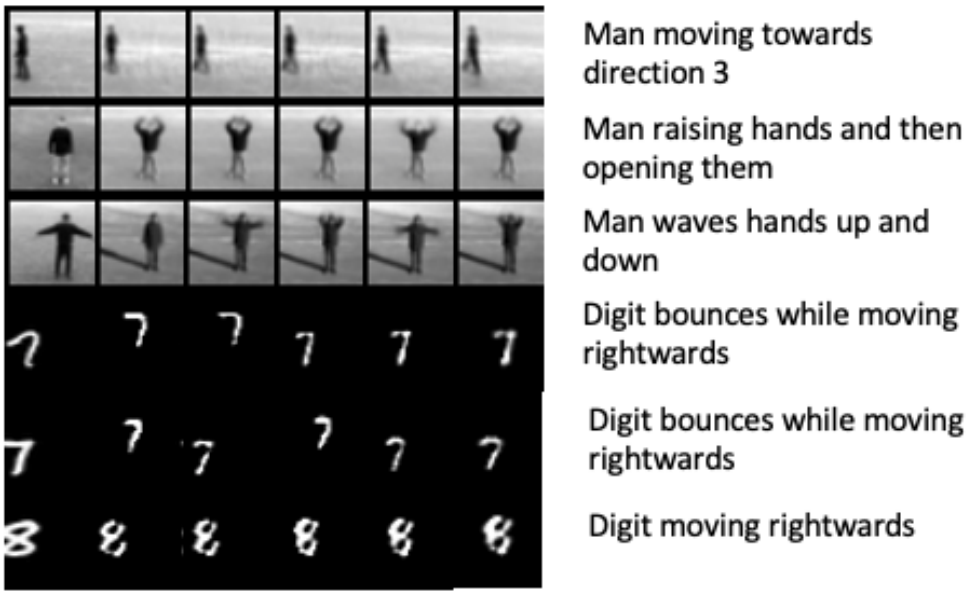


Figure 5: Experimental results related to experiments 2,3 from the main paper.

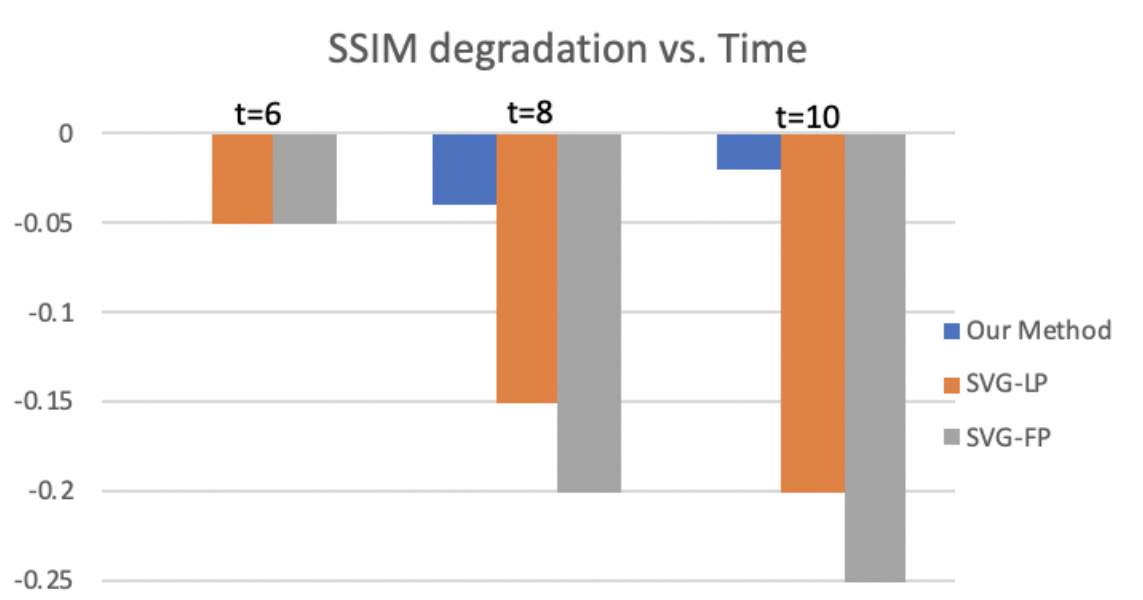


Figure 6: Comparison of the degradation of SSIM in our model against . Note that our model does not require 5 context frames (reason for starting to compare at frame 6) but only 1 and consists of only a simple 1-hidden-layer MLP compared to 70 convolutional layers in .

8 SSIM ANALYSIS

In Figure 6 we calculate the SSIM of our predicted frames against the ground truth and compare against Denton & Fergus (2018), we note that over the course of 10 time steps the quality of our frames does not degrade contrary to the autoregressive method of Denton & Fergus (2018). It is worth mentioning some crucial differences between the two approaches, Denton & Fergus (2018) uses a large architecture, $\sim$ 70 layers, and conditioning on at least five past frames. In contrast to this, our results were generated with a 1-hidden-layer MLP and only one previous frame is required to infer the future sampling path in the MST latent space.

REFERENCES

Emily Denton and Rob Fergus. Stochastic video generation with a learned prior, 2018.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. *ICML*, 2018a.

Octavian Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *NeurIPS*, 2018b.

Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. Adversarial autoencoders with constant-curvature latent manifolds. *Applied Soft Computing*, 2019.

Emile Mathieu, Charline Le Lan, Chris J. Maddison, Ryota Tomioka, and Yee Whye Teh. Continuous Hierarchical Representations with Poincaré Variational Auto-Encoders. *NeurIPS*, 2019.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in Neural Information Processing Systems*, (Nips), 2017.