We thank all reviewers for their constructive feedback and for their time in creating well thought out reviews.

Below we address all raised concerns, namely we perform ablation studies of adding (i) 2nd-order ODEs and (ii) BNNs; (iii) address more complex experiments and comparisons; and (iv) discuss the role of the KL and regularisation.

A new 1st-order baseline: We tested a new ODE^{1}VAE variant where the latent space is governed by 1st-order ODE system. ODE^{1}VAE is similar to the NeuralODE [Chen et al 2018], except for having BNNs, and for NeuralODE placing a variational distribution on initial value $q(\mathbf{x}_0)$, while ODE^{1}VAE models the posterior over full trajectory $q(\mathbf{x}_{0:T})$.

[R1,R3] ODE^{1}VAE vs ODE^{2}VAE: We performed a new comparison study of ODE^{1}VAE against ODE^{2}VAE on bouncing balls dataset. The experimental setup is kept the same, except that the number of convolutional filters is reduced so that the impact of differential function choice becomes more apparent. Table 1 shows the resulting MSE over 10 frame ahead predictions. Note that ODE^{2}VAE models the acceleration $\dot{\mathbf{v}}_t = \mathbf{f}(\mathbf{s}_t, \mathbf{v}_t) : \mathbb{R}^{2d} \to \mathbb{R}^d$ whereas 1st-order systems learn $\dot{\mathbf{z}}_t = \mathbf{f}(\mathbf{z}_t) : \mathbb{R}^d \to \mathbb{R}^d$. Results show that the 2nd-order dynamics results in far better accuracy, even if the first order dynamics has more flops ($d = 50$). We will include ablation studies in the paper.

Table 1: Comparison of neural network (NN) and Bayesian neural network (BNN) ODE's with different latent dimensionalities on BOUNCING BALL experiment. Adding 2nd order momentum achieves superior performance, while BNN's have a smaller impact.

	Latent dimensions d		Test MSE	
Model	1st-order state	2nd-order momentum	NN	BNN
ODE^{1}VAE	25	-	45	43
	50	-	36	35
ODE^{2}VAE	25	25	**26**	**27**

[R1,R2] NN vs BNN: Table 1 shows comparable performance of BNNs and NNs on bouncing balls. In order to demonstrate the benefit of using a BNN, we repeat the CMU walking experiment with a NN differential function. The MSE achieved by ODE^{2}VAE-NN over three test sequences is 9.96, whereas ODE^{2}VAE-BNN error improves to 9.43.

[R2] Learning of BNNs: Learning BNN is performed via mean-field variational approximation (simultaneously with variational inference of the whole ODE^{2}VAE model), where each weight and bias component has its own mean and shares a global variance parameter. The ODE solver used in our experiments is fixed step Runge-Kutta for both NN and BNN systems; hence NFEs are also the same.

[R1] Comprehensive experiments: Our model is suitable for sequential datasets, of which we demonstrated good performance on motion capture data, bouncing balls experiments and on rotating MNIST. Conventional image datasets such as CIFAR-10 or Celeb are not directly applicable for our model as they do not have an immediate dynamic dimension. In this work we proposed the theoretical foundations of latent differential equations, and in future we intend to explore video prediction application as separate work due to its daunting scope and complexity.

[R2] Comparison to moving MNIST: Moving MNIST is a dataset of digits bouncing off the walls of a box. Physical interaction rules in bouncing balls dataset is more complicated because balls collide with each other, as well. In that sense, inferring the dynamics in bouncing balls dataset is more challenging. On the other hand, MNIST dataset possibly requires more powerful decoders, which we will consider as part of future work on video prediction.

[R3] Missing NeuralODE baseline in rotating MNIST and bouncing balls: While the public NeuralODE implementation worked as expected in the CMU walking experiments, we were unable to get NeuralODE model to work in BOUNCING BALLS and ROTATING MNIST datasets. We included ConvNet architectures and tried these experiments numerous times with different encoder/decoder hyperparameters and initialisations; however we always got fully black frames as reconstructions. We believe the ODE^{1}VAE results instead to be informative enough to demonstrate inherent limitations of 1st-order models, such as NeuralODE.

[R1] Regularisation parameters: The β and γ parameters weigh the regularising KL terms to be comparable to the weight of the likelihood term (see e.g. "Fixing the Broken ELBO" paper). We choose to fix $\beta = |q|/|\mathbb{W}|$ to the ratio between the latent space dimensionality q and number of weight parameters of the differential function $|\mathbb{W}|$, in order to counter-balance the penalties. We chose $\gamma = 0.001$ by cross validation from [0,0.1,0.01,...0.00001].

[R2] ODE^{2}VAE-KL variant: As correctly pointed out by the reviewer, all consecutive triplets in a sequence are encoded. We then compute the KL divergence between encoder distributions and the state distributions induced by ODE integration. This way, the entire sequence (rather than only the initial values) is utilized for encoder training.

[R3] Long-term forecasting: Long-term forecasting of non-linear dynamical systems requires an almost perfect underlying dynamics model for the trajectories not to deviate. We regard "long-term" forecasting to be up around 20 frames ahead in bouncing balls, multiple cycles of walking, or a full rotation of MNIST numbers. We found out empirically that NeuralODE can not forecast sufficiently, while the GPPVAE model interpolates states over time with an RBF kernel with little extrapolation capability.