1. Additional Details.

1.1. Training details.

We fine-tune Lumiere on a dataset of 100K clips from Stereo4D as mentioned in section 4.3 of the main paper. We temporally subsample the videos into 80 frames at 16 fps to match Lumiere's pre-training temporal resolution. We train the model for 120K steps with batch size 32 and learning rate $2e^{-5}$. The original clips resolution is 512x512 pixels. To train the Eye2Eye base model, we additionally downsample the frames spatially to 128x128 pixels. For the Eye2Eye refiner, we randomly sample crops of 128 pixels.

1.2. Sampling hyper-parameters for our method.

1.2.1. Base Eye2Eye sampling.

We sample with 50 diffusion timesteps and without classifier free guidance. We sample from this model at a resolution of 256 pixels, as we found that this resolution best mitigates visual quality and 3D effect.

1.2.2. Eye2Eye refiner.

We upsample the output of the base Eye2Eye model 512x512 pixels resolution and noise it to diffusion timestep $t = 0.9$. We then denoise it with 48 diffusion timesteps and without classifier free guidance

2. Baselines.

2.1. Warp-and-inpaint implementation.

As warp-and-inpaint baselines [**?** **?**] do not have open source implementations, we implement and train this baseline. We use the same dataset described in **??** to fine tune the base Lumiere inpainting model to inpaint left-right disocclusion masks. We use [**?**] to estimate disparity of each pair of stereo frames, V_{left}, V_{right} and obtain the disocclusion mask by computing left-right consistency of the disparity prediciton. At training, the model is conditioned on the right video warped according to the estimated disparity, V_{right}^{warped}, and the corresponding disocclusion mask M, to denoise the left frame, with the standard diffusion objective:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t, V_{right}^{warped}, M)\|_2^2\right] \tag{1}$$

Here $x_t = V_{left} \cdot \sqrt{\alpha_t} + \sqrt{1-\alpha_t}\epsilon$, and $\epsilon \sim \mathcal{N}(0, I)$. At inference time, given a video V, we use SOTA monocular disparity estimation [**?**] to estimate video disparity D^V. As this estimation is scale and shift invariant, we fit a scale and shift parameter to the disparity map to align it with the disparity of our outputs (we first estimate the disparity of our outputs using [**?**]). We then forward-warp the frames using depth ordered softmax splatting [**?**]. The inpainting mask here are the pixel locations that were not mapped onto by D^V. We open and dilate the mask to reduce temporal inconsistencies before feeding it to the base inpainitng model. For spatial super resolution, we use the pretrained Lumiere SSR model and take a blended diffusion approach for maintaining faithfulness to the original video. Specifically, given the low resolution base model video output $V_{128\times128}^{inpainted}$, the warped right video $V^{\text{warped}} = \text{softmax\_z\_splatting}(V, D^v)$, we blend the predicted clean super-resolved output $\tilde{x_0}_t(x_{t+1}, t+1, V_{128\times128}^{inpainted})$ with V^{warped}: and at each diffusion step t $x_{0t} = M \cdot \tilde{x_0}_t(x_{t+1}, t+1, V_{128\times128}^{inpainted}) + (1-M) \cdot V^{warped}$. We use a the standard lumiere sampling of 256 and 32 diffusion timesteps for the base model and the SSR model, respectively, and a classifier free guidance of 8.

2.2. Deep3D.

As the original paper implementation uses a deprecated codebase, we turn to a more recent implementation found in the link: https://github.com/HypoX64/Deep3D. Their training data consists of 3D movies, which are typically processed in a differnt manner then our data - the zero disparity plane is usually shited to increase human comfort, making the RGB comparison difficult. We thus encourage the viewer to use anaglyph glasses for these results.

2.3. Dynamic Gaussian marbles.

We optimize the Dynamic Gaussian Marbles using the official paper implementation https://github.com/coltonstearns/dynamic-gaussian-marbles, using their default real-world videos configuration. We observed the optimizing the representation for the full number of steps (100K) in this configuration diverges, and thus synthesize stereo views from it after 40K steps.