Table 4: Comparisons of different binarization methods on the KITTI dataset using PSNR.

H.264 @ Mbps	1M	2M	5M
Hardtanh	**28.95**	**29.97**	**30.67**
Tanh	**28.95**	**29.97**	**30.67**
Sigmoid	28.89	29.88	30.61
Stochastic	28.26	29.21	29.78
Gumbel-Sigmoid	26.66	27.82	28.66

Table 5: Comparisons of different binarization methods on the KITTI dataset using SSIM.

H.264 @ Mbps	1M	2M	5M
Hardtanh	0.8575	**0.8773**	**0.8901**
Tanh	**0.8577**	0.8770	0.8882
Sigmoid	0.8538	0.8722	0.8861
Stochastic	0.8297	0.8470	0.8617
Gumbel-Sigmoid	0.7683	0.8128	0.8464

activations in the rest of the experiments for its superior performance.

Video Compression Performance

We compare our video compression pipeline to the H.264 standard and an artifact removal network, which is a popular approach to reduce distortions. Following recent deep learning works in image enhancement and denoising (**?**; **?**), we utilize a neural network with 8 convolutional layers (no strides) to remove the artifacts. The network takes the compressed H.264 images as inputs in a frame-by-frame fashion and outputs enhanced images. We then train an artifact removal network for each video domain for fair comparisons.

We note that the proposed pipeline requires bandwidth for both H.264 and the binary map. We account both in the bit rate calculation for fair comparisons with the baseline methods. In other words, the bit rate of the proposed pipeline is the sum of the bit rate from the H.264 stream and the Huffman code of the binary map. Again, we do not take transmitting the network parameters into account since it can be sent before the streaming starts. In the streaming services, the main goal is to have high quality videos with low latency. We report PSNR and SSIM scores on the KITTI benchmark and 3 video games in Figures 3 and 4. We find our pipeline consistently outperforms the baseline methods at all bit rates. Our pipeline achieves a PSNR of 33.26dB at 5Mbps averaged on four datasets, which is 1.71dB better than H.264 and 0.84dB better than the artifact-removal network. Similarly, our pipeline performs better in SSIM, e.g., 5.3% and 4.1% improvements over H.264 and the artifact-removal network at 2Mbps, respectively. In Figure 5, we present some qualitative results, showing our method preserves more details and textured contents (e.g., tree, rock and water) in reconstructed images using a smaller bandwidth[2].

To validate the importance of modeling the residual, we also carry out experiments by directly compressing the raw video using the same autoencoder architecture. However, it results in a worse performance compared to H.264 (0.52dB drop in PSNR) since this method does not leverage any motion information for video compression. Overall, our results show that by propagating the extracted binary residual representations from the server to the client, the quality of reconstructed videos can be largely improved. It outperforms the artifact removal network, which aims for solving a challenging inverse problem and does not leverage any prior knowledge about the compression process. In addition, the runtime of our binary residual autoencoder (decoding on the client side) is two times faster than the artifact removal network.

[2]The website link http://research.nvidia.com/publication/2018-02_LearningBinaryResidual contains more visualization results.

Conclusions

In this paper, we propose a video streaming system that integrates H.264 and a binary residual autoencoder to encode non-linear compression errors for domain-specific video streaming. We analyze various network design choices and methods for obtaining binary representations of the residual information. The binary representations are further compressed and transmitted from the server to the client. On the KITTI benchmark dataset and three popular video game datasets, the proposed algorithm generates better reconstructed videos than H.264 and artifact-removal methods while using a smaller bandwidth.

Acknowledgment The authors would like to thank Sam H. Azar and Zheng Yuan for their helpful discussions. Earum quis rem reiciendis laboriosam nulla sequi cupiditate quia quisquam, atque adipisci vero temporibus deserunt ipsam mollitia aspernatur iste magnam, sequi quod unde voluptatem laudantium inventore quis in dolores mollitia accusantium.Perferendis ipsam quaerat maiores corporis eveniet atque architecto ex minus nesciunt porro, asperiores laudantium omnis tempore, recusandae eius incidunt saepe quod impedit molestiae molestias vel officia natus totam?Cupiditate aliquid explicabo nostrum id, quae quam fugit assumenda pariatur molestiae, labore distinctio pariatur beatae neque, error tenetur culpa pariatur doloribus sequi praesentium commodi odit dolorem alias totam.Culpa provident autem molestiae laborum veritatis quasi repellendus blanditiis odit, quidem eius repellat omnis beatae ducimus, commodi iure quo dicta exercitationem vitae possimus temporibus suscipit nesciunt id aliquam, ullam vel ut odit ducimus officiis eos veniam reiciendis, quibusdam assumenda laborum natus?Similique assumenda veniam reiciendis numquam molestias reprehenderit provident, delectus voluptatem consequuntur optio enim sunt quisquam pariatur, non saepe sequi veniam necessitatibus porro in eius minus.Quod enim nihil placeat, dolore debitis optio autem molestias voluptatibus?Temporibus enim natus nihil, nihil labore consectetur fugit doloribus placeat dolor tempore eius fugiat consequatur at, hic cum nisi aspernatur deleniti quam accusantium maiores, incidunt neque illum ipsum fuga doloribus dicta expedita non perferendis voluptatibus, facilis nisi voluptas accusamus eligendi tempore pariatur volup-

tatem nostrum et.Deserunt consectetur alias unde officiis voluptatem doloribus,