A. Additional Qualitative Results

We showcase additional qualitative results comparing our stereo compression model against other baselines, from BPG and JPEG2000 to single-image Ballé to our residual coding baseline. We note that for the baselines, we report a separate bitrate per camera, since each camera image is compressed as a single image. However, for our stereo model, we report the joint bitrate divided by 2. The reason for this is that even though our models outputs a separate code for each image, the first code $\bar{\boldsymbol{y}}_1$ may contain additional information to help the compression of the second code $\bar{\boldsymbol{y}}_2$, since it is used as an input for both our skip functions and conditional entropy. We report separate perceptual metrics per camera image for all models.

A.1. Additional Qualitative Results from Stereo Model

Here, we showcase additional qualitative results for Cityscapes and NorthAmerica in Fig. 6 and 8 respectively. On Cityscapes, where the image resolution is 1920×720, the closest competing algorithm is generally BPG. We highlight the differences and tradeoffs between our model and BPG in Fig. 6. In general, BPG tends to do a better job at preserving certain high-frequency information - such as road signs and license plates - at the expense of introducing artifacts, discoloration, and blurriness in other regions. We also find BPG might enhance some high-frequency regions while blurring others. In contrast, our model provides a more consistent level of detail across all image regions.

On NorthAmerica, where the image resolution is 480×300, our model demonstrates more crisp results at lower bitrates compared to all competing algorithms, as shown in Fig. 8.

A.2. Artifacts from Residual Coding Baseline

We showcase results from our residual coding baseline. In Fig. 11, we compare the reconstructions between camera 1 (effectively produced via a single-image Ballé network), and camera 2 (produced via the output of motion-compensation using SGM and residual coding using a separate Ballé network) on a stereo image pair in Cityscapes. We additionally include the same output from our stereo model. The camera 2 reconstruction has overall higher perceptual metrics in terms of PSNR/MS-SSIM at a lower bitrate, and also that certain regions in the image look undeniably sharper than in camera 1 and in the outputs from our own stereo model (shown by the green boxes). However, we highlight other regions, shown by the red boxes, where there exist jarring artifacts in the camera 2 reconstruction that are not present in camera 1 nor in our stereo model outputs. There are cuts/tears around the boundaries where SGM does not output valid disparities; moreover there exist significant warping artifacts around regions with larger disparities that are predicted less accurately.

We did not attempt any additional fine-tuning or refinement after merging the residual image with the disparity-warped first reconstruction to construct the second reconstruction. We leave that as an interesting direction to explore in future work. We also note that the artifacts start to go away at higher bitrates, but at that point the overall performance of the stereo residual baseline also deterioriates to below the curve of the single-image Ballé model.

The artifacts also exist in NorthAmerica, where our deep residual coding baseline underperforms even single-image compression at all bitrates. We show a sample original/reconstructed disparity map, residual image, and final image in Fig. 10.

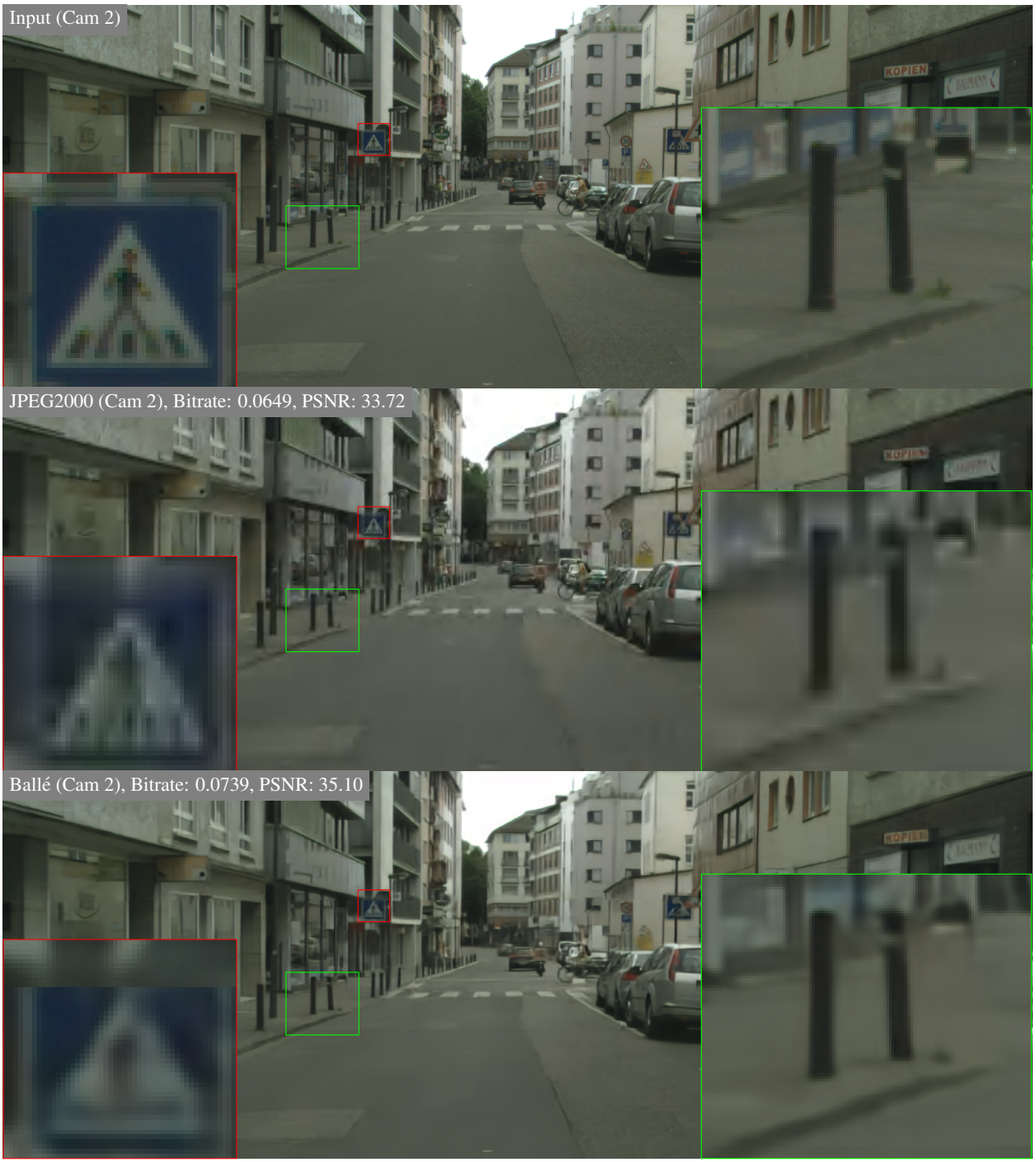

Figure 6: Comparison between the reconstructions of competing baselines and our method on a Cityscapes image (in camera 2). We focus on comparing our method with BPG (next page). BPG sharpens high-frequency details and distorts others with blurriness and/or artifacts, whereas we provide a more consistent level of detail across the image. The red box demonstrates a region where BPG provides shaper details, whereas the green box demonstrates where our method is better.

B. Expanded Ablation Studies

In our ablation study in the main paper (see Fig 5. in the main paper), we measure the independent effects of our parametric skip functions, conditional entropy, and hyperprior by adding them on top of a factorized-prior model. Here, in our expanded ablation study, we measure

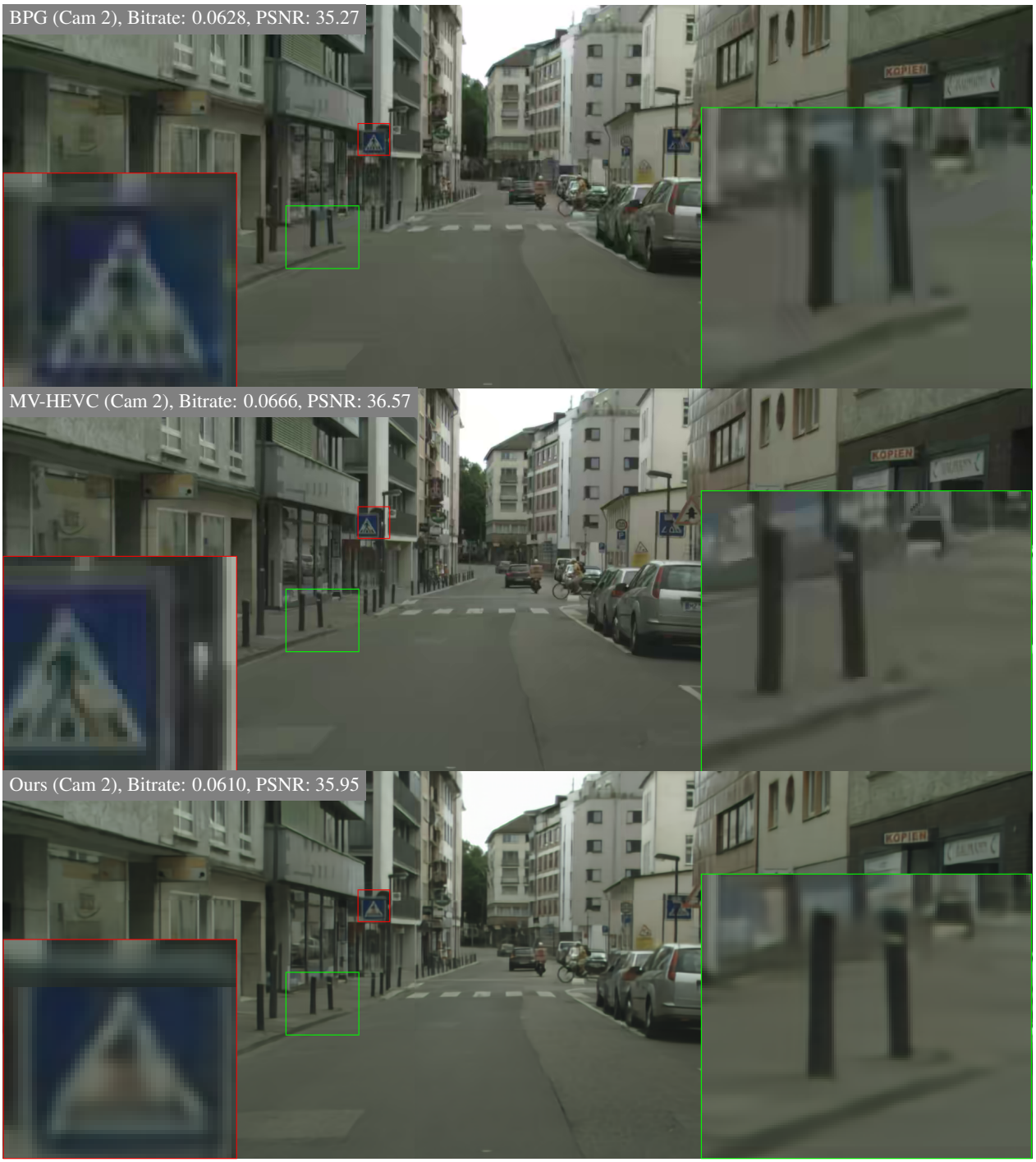

Figure 6: (Continued)

the separate effects of our parametric skip functions and conditional entropy when combined with a single-image hyperprior model.

The results are shown in Fig. 12. We observe that our results follow a similar trend to that in the main paper. DispSkip provides the highest bitrate savings and perceptual metrics gains at lower bitrates, and decreases for higher bitrates (especially for Cityscapes). Meanwhile, adding a conditional entropy component adds relatively consistent bitrate savings at all levels compared to the hyperprior model. We also observe cannibalization effects when combining DispSkip with conditional entropy, which

Figure 7: Additional qualitative comparions on a Cityscapes image (in camera 1).

is also observed in the main paper; yet again, combining DispSkip with conditional entropy yields the best results at all bitrates.

C. Additional Architecture Details

We provide additional architecture details in this section on various aspects of our model. First, in Section C.1, we provide some more details about our main encoder/decoder architecture. Then, in Section C.2, we provide architecture

Figure 7: (Continued)

details about the main components of our parametric skip functions: predicting the global context, predicting the cost volume at each level of the encoder/decoder, as well as the final feature aggregation. Finally, in Section C.3, we provide details for the varous components that make up our conditional entropy model: our hyper-encoder (deriving hyperpriors from our image code), our factorized prior entropy model for our hyperpriors, and our GMM-based model for our image codes.

Figure 8: Comparison between the reconstructions of competing baselines and our method on a NorthAmerica stereo pair. We observe that our method yields the highest PSNR at the lowest bitrate compared to all competing methods (34% reduction in residual bitrate compared to Ballé).

C.1. Additional Architecture Details for Encoder/Decoder

The number of channels for each intermediate layer in both the encoder/decoder of each image is set to N, and the number of channels of each of the two codes, $\bar{\boldsymbol{y}}_1$, $\bar{\boldsymbol{y}}_2$ is set to M. For the lower bitrates (< 0.7), we set $N = 100$ and $M = 140$; we found that setting a smaller bottleneck didn't affect model performance too much and allowed the models to train much faster. For the higher bitrates (≥ 0.7), we set $N = 192$ and $M = 256$.

C.2. Architecture Details of Parametric Skip Function

Recall that our parametric skip functions consist of four main components. A **global context** feature is predicted from the code of image 1 $\bar{\boldsymbol{y}}_1$, in order to capture global information from image 1. Then, at each level of the

Figure 8: (Continued)

encoder/decoder, we predict a **stereo cost volume** from $\boldsymbol{h}_1^{t-1}$, $\boldsymbol{h}_2^{t-1}$ - the feature maps of image 1 and 2 from the previous layer - as well as the global context feature. We use the cost volume to **densely warp** $\boldsymbol{h}_1^{t-1}$ from image 1 to image 2, and finally **aggregate** this warped feature with $\boldsymbol{h}_2^{t-1}$. We describe the architecture details of predicting the global context, predicting the stereo cost volume at each level, and aggregating the features below.

Global Context: The global context module takes as input $\bar{\boldsymbol{y}}_1$, the first image code, with dimensions $M \times H/16 \times W/16$, where M is the channel dimension and H, W are the height/width of the original image. It passes $\bar{\boldsymbol{y}}_1$ through four 2D convolutional layers. Each conv layer except the last is followed by a GroupNorm [50] and ReLU layer. In general we use GroupNorm instead of BatchNorm [18] in our models due to our small batch sizes.

The dimension of each intermediate feature is $F \cdot C$, where C is our maximum disparity and F is a multiplicative factor. The final global context output after

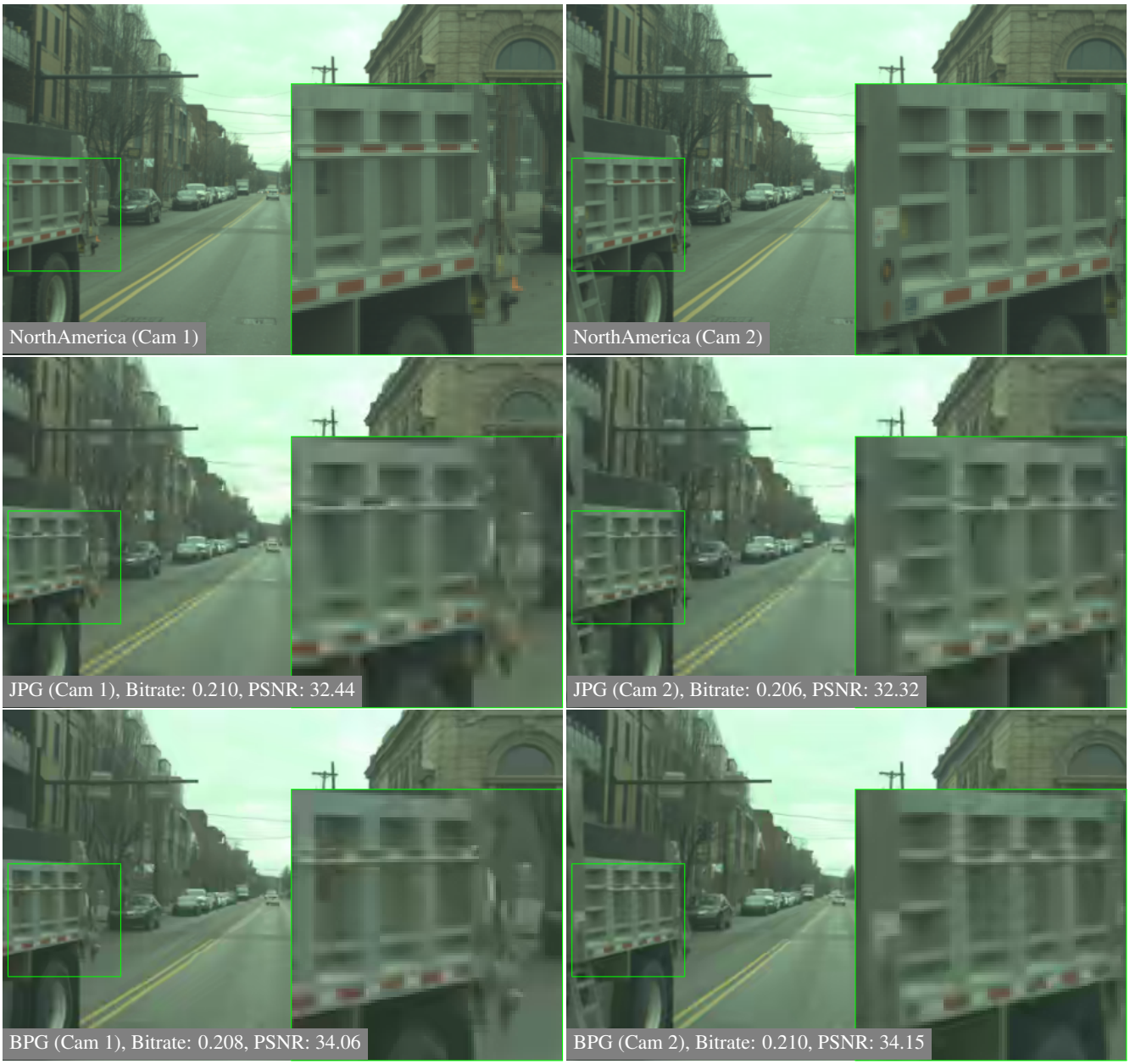

Figure 9: Additional qualitative comparison on a NorthAmerica stereo pair. We observe a 46% reduction in residual bitrate compared to Ballé with higher PSNR.

the convolutional layer is $(F \cdot C) \times H/16 \times W/16$, which we reshape into a 4D volume: $F \times C \times H/16 \times W/16$. Hence our global context can be seen as an initial cost volume (with an additional feature dimension), which we will provide as input to our skip functions at each level of our encoder/decoder.

Note that we have three levels of skip functions in both the encoder/decoder, predicting cost volumes of dimensions $C \times H/2 \times W/2$, $C \times H/4 \times W/4$, and $C \times H/8 \times W/8$ for the encoder and of dimensions $C \times H/8 \times W/8$, $C \times H/4 \times W/4$, and $C \times H/2 \times W/2$ for the decoder. Since the disparity dimension remains fixed regardless of spatial resolution, the lower resolution cost volumes effectively have a greater receptive field than the higher resolution volumes (ideally we would like the higher resolution volumes to have a big receptive field but this is subject to GPU memory limits). This also implies that the disparity dimensions are not spatially aligned across different spatial resolutions nor with our global context (at the lowest spatial resolution $H/16 \times W/16$), so feeding our

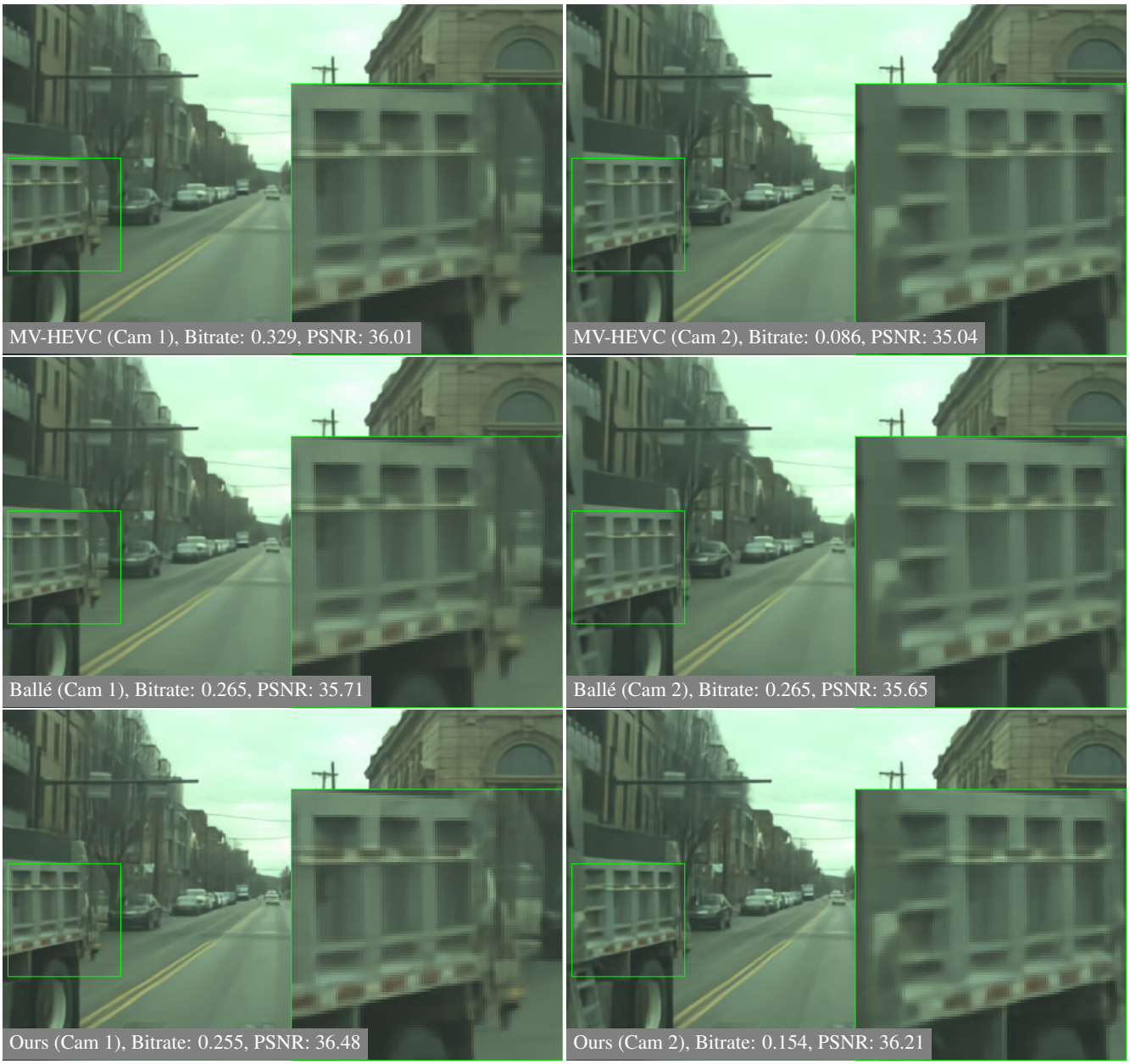

Figure 9: (Continued)

global context as is to each level doesn't make sense.

Instead, we ensure that F is divisible by 3, and our global context volume actually represents a concatenation of three "sub" context volumes of dimensions $F_0 \times C \times H/16 \times W/16$, where $F_0 = F/3$. Each sub-context volume is mapped as an input to a skip function at a corresponding resolution level in both the encoder/decoder (so one sub-context volume is mapped to the skip function in both the encoder and decoder at resolution $H/8, W/8$, etc.). This allows each sub-context volume to represent a lower-resolution feature representation to help predict a specific cost volume at a particular resolution level, as opposed to helping predict all cost volumes across all resolution levels.

A network diagram is shown in Fig. 13. We set $F = 21$ in our experiments. As mentioned in our experiments, we set $C = 32$ for NorthAmerica and $C = 64$ for Cityscapes. We set GroupNorm to have F groups, with C channels per group.

Stereo Cost Volume If the input features to each skip function are at level $t - 1$ with resolution r, denote the

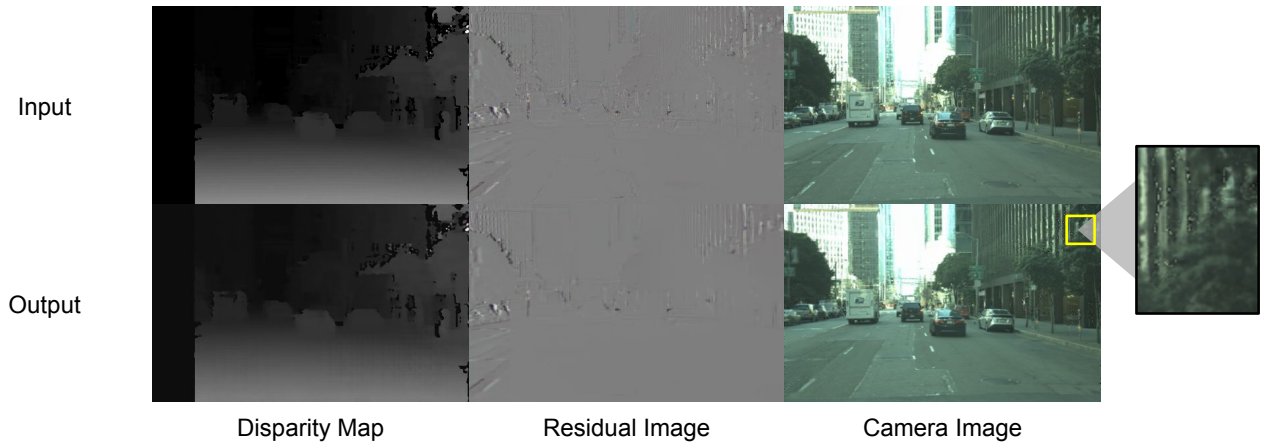

Figure 10: Input/Output disparity map, residual image, and camera image for a sample stereo pair on NorthAmerica (using our stereo residual coding baseline).

corresponding sub-context volume from the global context as $\boldsymbol{d}^r$. The task of predicting the cost volume used for warping takes in $\boldsymbol{d}^r$, as well as $\boldsymbol{h}_1^{t-1}$, $\boldsymbol{h}_2^{t-1}$ as input.

We concatenate $\boldsymbol{h}_1^{t-1}$, $\boldsymbol{h}_2^{t-1}$ into a $2N \times H^{t-1} \times W^{t-1}$ feature, and feed it through 2 2d convolutions, followed by GroupNorm (with 4 groups per module) and ReLU after each conv. The output feature has dimensions $N \times H^{t-1} \times W^{t-1}$.

In another branch, we feed $\boldsymbol{d}^r$, the sub-context volume, through an upsampling 3d conv. to match the spatial resolution of $\boldsymbol{h}_1^{t-1}$, $\boldsymbol{h}_2^{t-1}$ (which is H^{t-1}, W^{t-1}), followed by another 3d conv. Each 3d conv is also followed by GroupNorm (1 group per module) and ReLU, and the intermediate feature channel dimensions are $C \cdot F_0$. The output feature has dimensions $F_0 \times C \times H^{t-1} \times W^{t-1}$, and we collapse this back into a 2d feature representation: $(F_0 \cdot C) \times H^{t-1} \times W^{t-1}$.

We concatenate the outputs of both feature branches and add 3 more 2d conv layers, with intermediate feature dimension N, each except the last followed by GroupNorm(4 groups each) and ReLU. The final cost volume has dimensions $C \times H^{t-1} \times W^{t-1}$, with a softmax layer applied over the disparity dimension for every $0 \leq i, j \leq H^{t-1}, W^{t-1}$.

A network diagram for predicting the cost volume is given in Fig. 14.

Aggregation Function Our aggregation function $\boldsymbol{h}_2^t = a(\boldsymbol{g}_2^{t-1}, \boldsymbol{h}_2^{t-1})$ is fairly simple - since $\boldsymbol{g}_2^{t-1}$ and $\boldsymbol{h}_2^{t-1}$ have the same spatial resolution, we concatenate them along the channel dimension. Then we apply a downsampling/upsampling conv as part of the second image's encoder/decoder, as shown in Fig. 1 of the main paper.

C.3. Architecture Details for Entropy Models

Hyper-encoder: Our "hyper-encoder" derives the hyperprior variables, $\bar{\boldsymbol{z}}_1, \bar{\boldsymbol{z}}_2$ from $\boldsymbol{y}_1, \boldsymbol{y}_2$. Note that we pass the unquantized continuous representation $\boldsymbol{y}$ into the hyper-encoder, not $\bar{\boldsymbol{y}}$, the noisy representation produced by the quantizer during training. Each $\boldsymbol{y}$ is fed through 3 convolution layers, with ReLUs following the first two and the last two being downampling; then a quantizer is applied to produce $\bar{\boldsymbol{z}}$. An illustration can be shown in Fig. 15.

Hyperprior Entropy Model: We follow [5] in designing the factorized entropy model for the hyperprior - specifically in modeling $c_i(\bar{z}_i; \boldsymbol{\theta}_{\bar{\boldsymbol{z}}})$. In order to define a valid cumulative density, $c_i(\bar{z}_i; \boldsymbol{\theta}_{\bar{\boldsymbol{z}}})$ must map values between $[0, 1]$ and be monotonically increasing. The input $\bar{z}_i$ and the output must also be univariate (dimension = 1).

We set c_i to be a two-step nonlinear function as follows:

$$c_i(\bar{z}_i; \boldsymbol{\theta}_{\bar{\boldsymbol{z}}}) = f_2 \circ f_1 \tag{17}$$

where $f_1 : \mathbb{R}^1 \to \mathbb{R}^3$ and $f_2 : \mathbb{R}^3 \to \mathbb{R}^1$. The nature of each f_k is defined as follows:

$$\begin{aligned} f_k(\boldsymbol{x}) &= g_k(\text{softplus}(\boldsymbol{H}^k)\boldsymbol{x} + \boldsymbol{b}^k) \\ g_1(\boldsymbol{x}) &= \boldsymbol{x} + \tanh(\boldsymbol{a}^k) \odot \tanh(\boldsymbol{x}) \\ g_2(\boldsymbol{x}) &= \text{sigmoid}(\boldsymbol{x}) \end{aligned} \tag{18}$$

where $\boldsymbol{H}^k$ are matrices, $\boldsymbol{b}^k$ and $\boldsymbol{a}^k$ are vectors, and $\odot$ is elementwise multiplication. This formulation satisfies the conditions to be a valid CDF. For more details and justifications about this manner of designing a factorized prior, see Appendix 6.1 in [5].

We use this same factorized prior formulation for modeling our main image codes in our models without

Figure 11: Comparison between the reconstructions from the two cameras using our deep residual coding baseline as well as our stereo model, for a Cityscapes stereo pair. The green box demonstrates where our residual baseline reconstruction (Cam 2) has sharper image quality than our stereo model. The red boxes demonstrate where the residual baseline reconstruction (Cam 2) introduces artifacts that are absent in our stereo model.

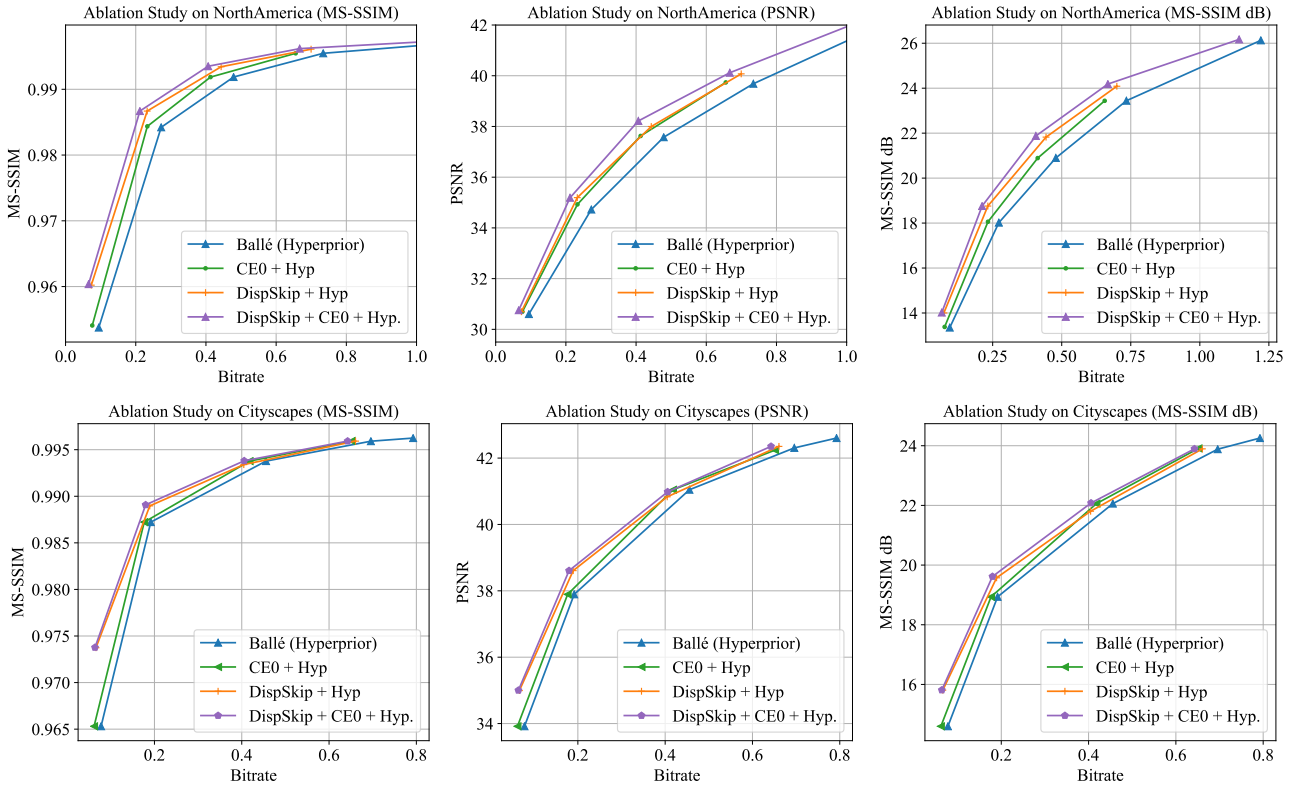

Figure 12: Additional ablation study. For both datasets, we analyze the independent and combined effects of our skip functions (DispSkip) and the conditional entropy on top of the single-image hyperprior model.

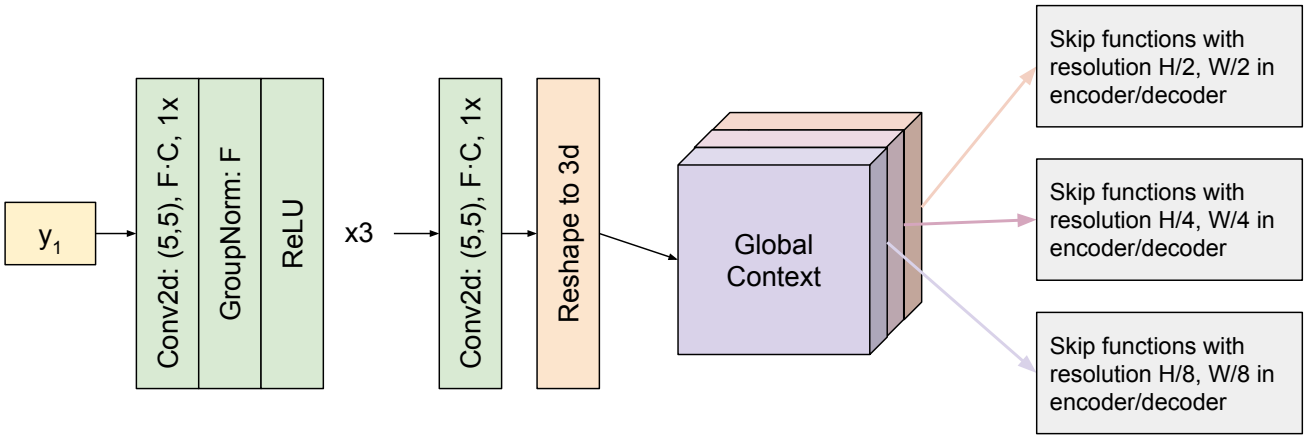

Figure 13: Architecture diagram for producing the global context volume from $\bar{\boldsymbol{y}}_1$, with 3 subcontexts. Each subcontext is passed to the two corresponding skip functions at that resolution level, one in the encoder and one in the decoder.

hyperpriors in our ablation study (Section 4.3 in the main paper). For our IE models, we used the factorized prior model for both image codes. For our CE0 models, we used this factorized prior model for the first image code.

Image Codes Entropy Model: We now describe the GMM-based conditional entropy model for the image codes: $\bar{\boldsymbol{y}}_1, \bar{\boldsymbol{y}}_2$. We start with $\bar{\boldsymbol{y}}_1$. Recall that we define $p_{1,i}(\bar{y}_{1,i}|\bar{\boldsymbol{z}}_1; \boldsymbol{\theta}_{\bar{\boldsymbol{y}}_1}) = (q_{1,i} * u)(\bar{y}_{1,i})$, where $q_{1,i} = \sum_k w_{ik}\mathcal{N}(\mu_{ik}, \sigma_{ik}^2)$. We predict $\boldsymbol{w}$, $\boldsymbol{\mu}$, and $\boldsymbol{\sigma}$ as functions

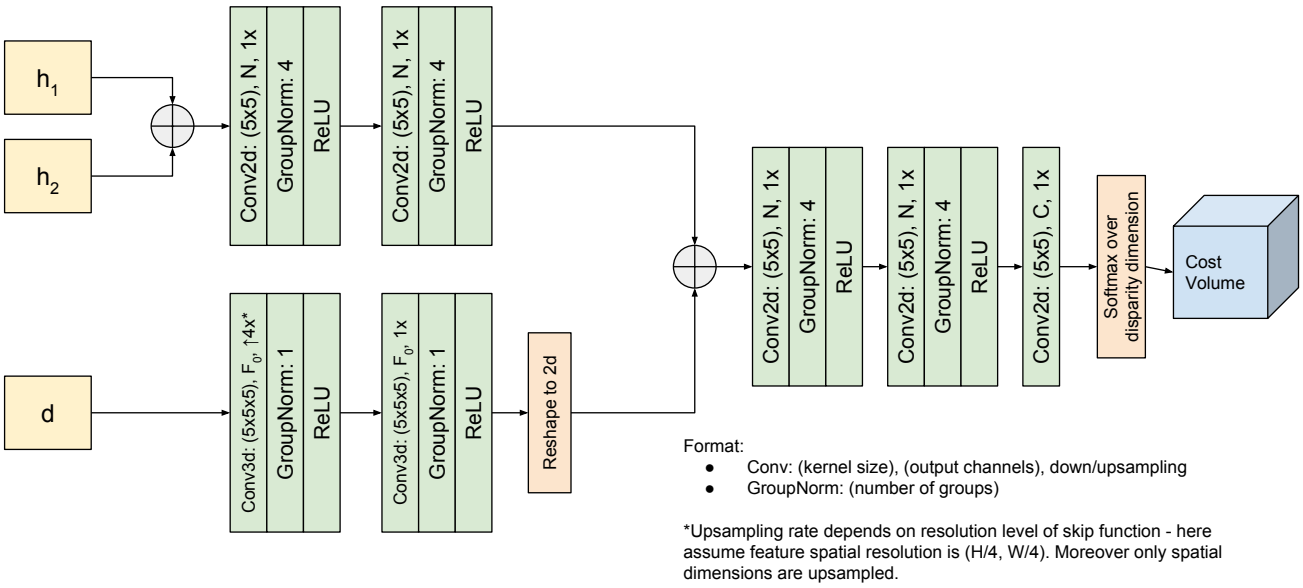

Figure 14: Architecture diagram for producing the cost volume from $\boldsymbol{h}_1^{t-1}$, $\boldsymbol{h}_2^{t-1}$, $\boldsymbol{d}^r$. The $\oplus$ symbol represents concatenating two tensors along the channel dimension.

of $\boldsymbol{z}_1$ given $\boldsymbol{\theta}_{\bar{\boldsymbol{y}}_1}$: $w(\bar{\boldsymbol{z}}_1;\boldsymbol{\theta}_{\bar{\boldsymbol{y}}_1})$, $\mu(\bar{\boldsymbol{z}}_1;\boldsymbol{\theta}_{\bar{\boldsymbol{y}}_1})$, $\sigma(\bar{\boldsymbol{z}}_1;\boldsymbol{\theta}_{\bar{\boldsymbol{y}}_1})$ - where $\boldsymbol{w}$, $\boldsymbol{\mu}$, and $\boldsymbol{\sigma}$ represent the vectors of all the individual values $w_{ik}, \mu_{ik}, \sigma_{ik}$. $\boldsymbol{\sigma}$ and $\boldsymbol{\mu}$ have the same spatial resolution as $\bar{\boldsymbol{y}}_1$ with up to K times the number of channels, where K is the number of mixtures ($(M \cdot K) \times H/16 \times W/16$). Moreover, to reduce the number of parameters and help maintain spatial invariance, we assume that weights are fixed per channel, so weights have dimensions $(M \cdot K) \times 1 \times 1$. The network diagram is shown in Fig. 15, and there are a few key details per branch. Namely, we apply a ReLU to the last layer of $\sigma(\bar{\boldsymbol{z}}_1;\boldsymbol{\theta}_{\bar{\boldsymbol{y}}_1})$ to keep standard deviations positive. For weights, we apply a pooling layer after the second conv to collapse the spatial dimension, then a softmax per mixture to keep weights normalized.

We follow a similar process to model $p_{2,i}(\bar{y}_{1,i}|\bar{\boldsymbol{z}}_2, \bar{\boldsymbol{y}}_1; \boldsymbol{\theta}_{\bar{\boldsymbol{y}}_2}) = (q_{2,i} * u)(\bar{y}_{2,i})$. However, the network structure for predicting $\boldsymbol{w}$, $\boldsymbol{\mu}$, and $\boldsymbol{\sigma}$ is slightly different because $\bar{\boldsymbol{z}}_2, \bar{\boldsymbol{y}}_1$ are not the same dimension. Instead, we first upsample $\bar{\boldsymbol{z}}_2$ to an intermediate value with the same dimensions of $\bar{\boldsymbol{y}}_1$. Then we can concatenate this intermediate value with $\bar{\boldsymbol{y}}_1$ across the channel dimension and pass it through the convolutions. The convolutions themselves are no longer upsampling, since the input is at the same desired spatial resolution as the output. An example for predicting $\boldsymbol{\sigma}$ is shown in Fig. 16.

D. Effect of Different Lossless Coders:

For lossless encoding, we compare our range coding [29] implementation against Huffman coding and zlib [26]. We find that range coding achieves a bitrate that is within 1-2% of the Shannon entropy lower bound. As a comparison, our Huffman coding implementation with a tuned chunk size uses 35-50% more bits than the Shannon entropy. Finally, the DEFLATE algorithm used in zlib (a combination of LZ77 and Huffman) uses between 150%-200% more bits.

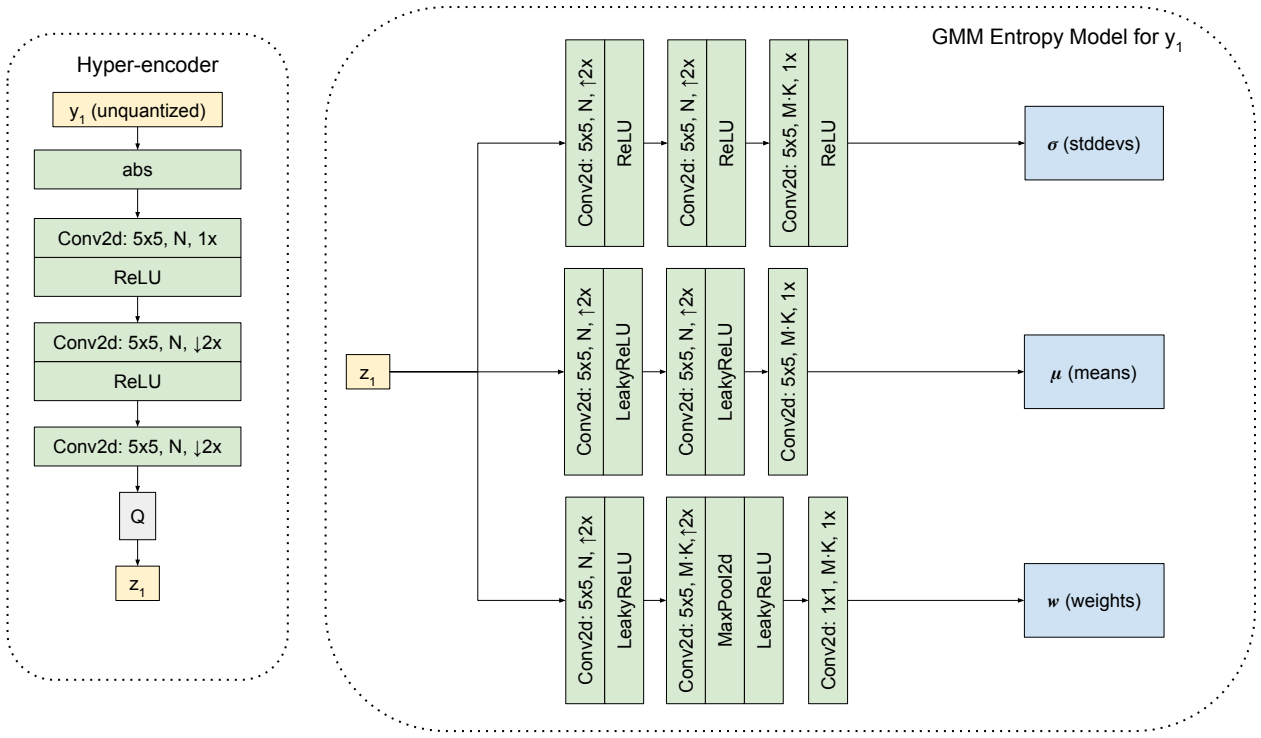

Figure 15: Architecture diagram detailing the hyper-encoder (left) as well as the full entropy model of $\bar{\boldsymbol{y}}_1$ (right). We note that the input to the hyper-encoder is $\boldsymbol{y}_1$ (the continuous representation before being fed to the quantizer), not $\bar{\boldsymbol{y}}_1$ (the noisy representation of $\boldsymbol{y}_1$ we apply as part of the quantizer during training). The hyperencoder produces $\bar{\boldsymbol{z}}_1$, which we then feed into the GMM entropy model.

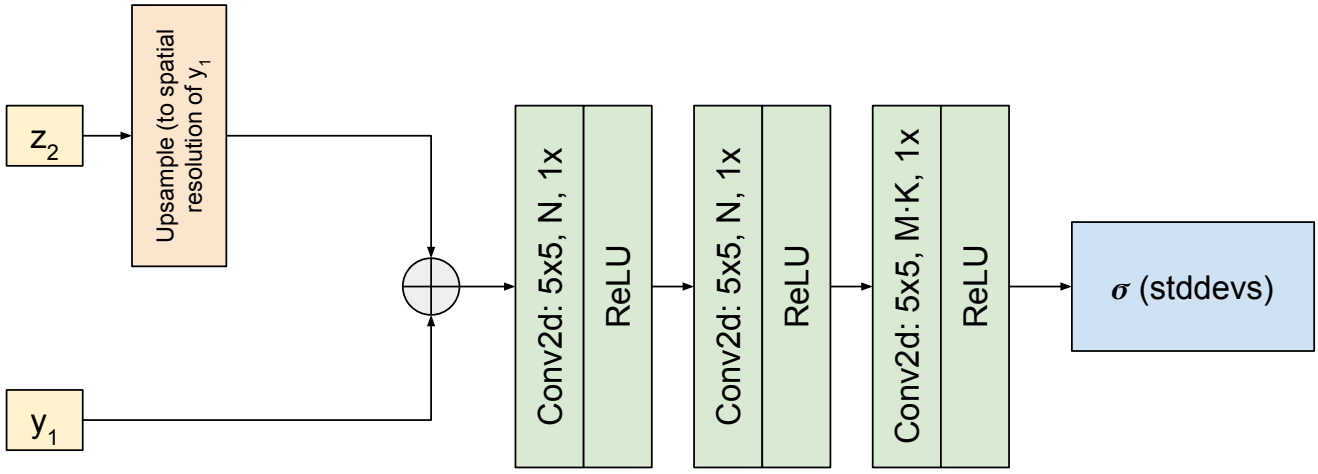

Figure 16: Architecture diagram illustrating how $\boldsymbol{\sigma}$ is predicted for $\bar{\boldsymbol{y}}_2$. The key difference is that $\bar{\boldsymbol{y}}_1$ is concatenated with an upsampled $\bar{\boldsymbol{z}}_2$ and the convolutions are no longer upsampling. The changes to predict $\boldsymbol{\mu}$, $\boldsymbol{w}$ are the same.