A. Supplementary Material – Multi-Realism Image Compression with a Conditional Generator

A.1. Runtime Benchmarks

We present runtime benchmark results on three systems. The first system used an NVIDIA Tesla V100, released in late 2017. The results for this system are presented in Table 1. The second system employs an NVIDIA 2080 Ti, a consumer GPU targeted primarily at video gaming and released 2018. The results for this system are summarized in Table 2. The third system uses a more powerful NVIDIA 3090 Ti GPU, summarized in Table 3.

To present a realistic usage of the method, we include the CPU runtime needed to encode images (i.e., range coding is included in the total compression/entropy encoding/decoding/total decompression time).

We note that comparing runtime numbers across papers is challenging due to implementation details and platform-specific optimizations. In our case, we did not aim to provide the fastest runtime numbers and, as such, when implementing the ELIC method [13], we omitted some performance critical features. Primarily, we did not use checkerboard encoding, nor did we employ uneven grouping for CHARM. Both of these should provide better runtime performance for the entropy coding and decoding components.

A.2. Running Published methods

HiFiC We use the models published in Tensorflow Compression [4], using `tfci` to run the models called `hific-lo`, `hific-mi`, `hific-hi`

VTM We use VTM 17.1 [11]. To compress images, we convert them to YCbCr, store the result as raw bytes in at `$YUVPATH` and then run the following:

```
# Encode
EncoderApp -c encoder_intra_vtm.cfg
 -i $YUVPATH -q $Q, -o /dev/null
 -b $OUTPUT
 --SourceWidth=$WIDTH
 --SourceHeight=$HEIGHT
 --FrameRate=1 --FramesToBeEncoded=1
 --InputBitDepth=8
 --InputChromaFormat=444
 --ConformanceWindowMode=1

# Decode
DecoderApp
 -b $OUTPUT -o $RECON -d 8
```

BPG We use BPG v0.9.8 and run the following given input PNGs:

```
# Encode
bpgenc -o $OUTPUT -q $Q
  -f 444 -e x265 -b 8 $SRC

# Decode
bpgdec -o $RECON $OUTPUT
```

A.3. Comparison to Iwai et al.

To compare with [18], we ran their released model on CLIC and evaluated their model in terms of FID and PSNR versus bitrate, see Figure 8.

A.4. Comparing to ELIC Reconstructions

We compare to the two published ELIC reconstructions in Fig. 9.

A.5. More Realism rates

We show Fig. 4 at more rate poitns in Fig. 10.

A.6. Comparing conditioning schemes

In Figure 11 we compare the FourierCond and TableCond conditioning approaches. Both methods were trained in the same manner as our main model.

A.7. Visual Results

We provide CLIC 2020 reconstructions for our lowest rate model, both for $\beta = 0.0$ and $\beta = 2.56$ in a zip file which have uploaded here: https://storage.googleapis.com/multi-realism-paper/multi_realism_paper_supplement.zip

Additionally, we include the MD5 checksum of the file to allow for verifying its integrity:

```
$ md5sum \
 multi_realism_paper_supplement.zip

2ed64ff793bbb87b7a884750ab61c912
 multi_realism_paper_supplement.zip
```

Model	Encoder [ms]	Entropy Coding [ms]	Entropy Decoding [ms]	Decoder [ms]	Total Compression [ms]	Total Compression [mp/s]	Total Decompression [ms]	Total Decompression [mp/s]
Ours (FourierCond)	121.3	55.6	50.6	153.8	176.8	11.2	204.4	9.7
MSE Baseline (N=192)	71.2	52.3	47.6	86.3	123.6	16.0	133.9	14.8
SOTA MSE Baseline (N=256)	121.4	66.6	74.3	139.0	188.0	10.5	213.3	9.3
HiFiC	43.2	47.4	57.4	200.5	90.6	21.9	257.9	7.7

Table 1. Runtime numbers (ms) needed to process a one megapixel image on a **Tesla V100** using float32 precision. Compression/decompression numbers include predicting entropy parameters on the GPU and running the range coder/decoder on CPU.

Model	Encoder [ms]	Entropy Coding [ms]	Entropy Decoding [ms]	Decoder [ms]	Total Compression [ms]	Total Compression [mp/s]	Total Decompression [ms]	Total Decompression [mp/s]
Ours (FourierCond)	128.4	92.3	75.2	205.9	220.7	9.0	281.1	7.1
MSE Baseline (N=192)	81.4	89.1	72.4	109.2	170.5	11.6	181.6	10.9
SOTA MSE Baseline (N=256)	128.7	68.5	72.9	155.5	197.2	10.1	228.4	8.7
HiFiC	40.5	48.9	57.5	248.3	89.4	22.2	305.8	6.5

Table 2. Runtime numbers (ms) needed to process a one megapixel image on a **NVIDIA 2080 Ti** consumer GPU using float32 precision. Compression/decompression numbers include predicting entropy parameters on the GPU and running the range coder/decoder on CPU.

Model	Encoder [ms]	Entropy Coding [ms]	Entropy Decoding [ms]	Decoder [ms]	Total Compression [ms]	Total Compression [mp/s]	Total Decompression [ms]	Total Decompression [mp/s]
Ours (FourierCond)	62.7	45.9	37.2	80.7	108.6	18.3	117.9	16.8
MSE Baseline (N=192)	44.6	46.0	37.5	47.9	90.6	21.9	85.4	23.2
SOTA MSE Baseline (N=256)	62.9	47.2	54.0	68.8	110.1	18.0	122.8	16.1
HiFiC	20.7	33.2	46.8	109.7	54.0	36.7	156.5	12.7

Table 3. Runtime numbers (ms) needed to process a one megapixel image on a **NVIDIA 3090 Ti** consumer GPU using float32 precision. Compression/decompression numbers include predicting entropy parameters on the GPU and running the range coder/decoder on CPU. Please note that certain components don't scale linearly when compared to the **NVIDIA 2080 Ti**. This is because entropy encoding/decoding is still serial and the autoregressive components aren't parallelized.

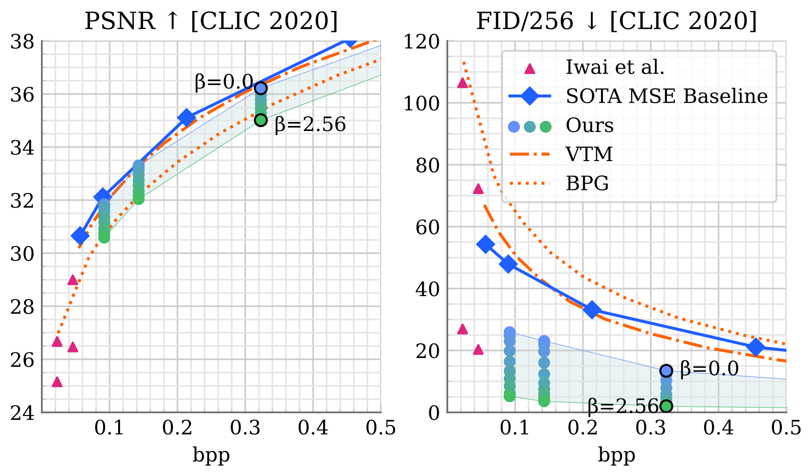

Figure 8. Comparison to Iwai et al. [18]. We can see that they target lower bitrates but this results in significantly worse PSNR and FID.

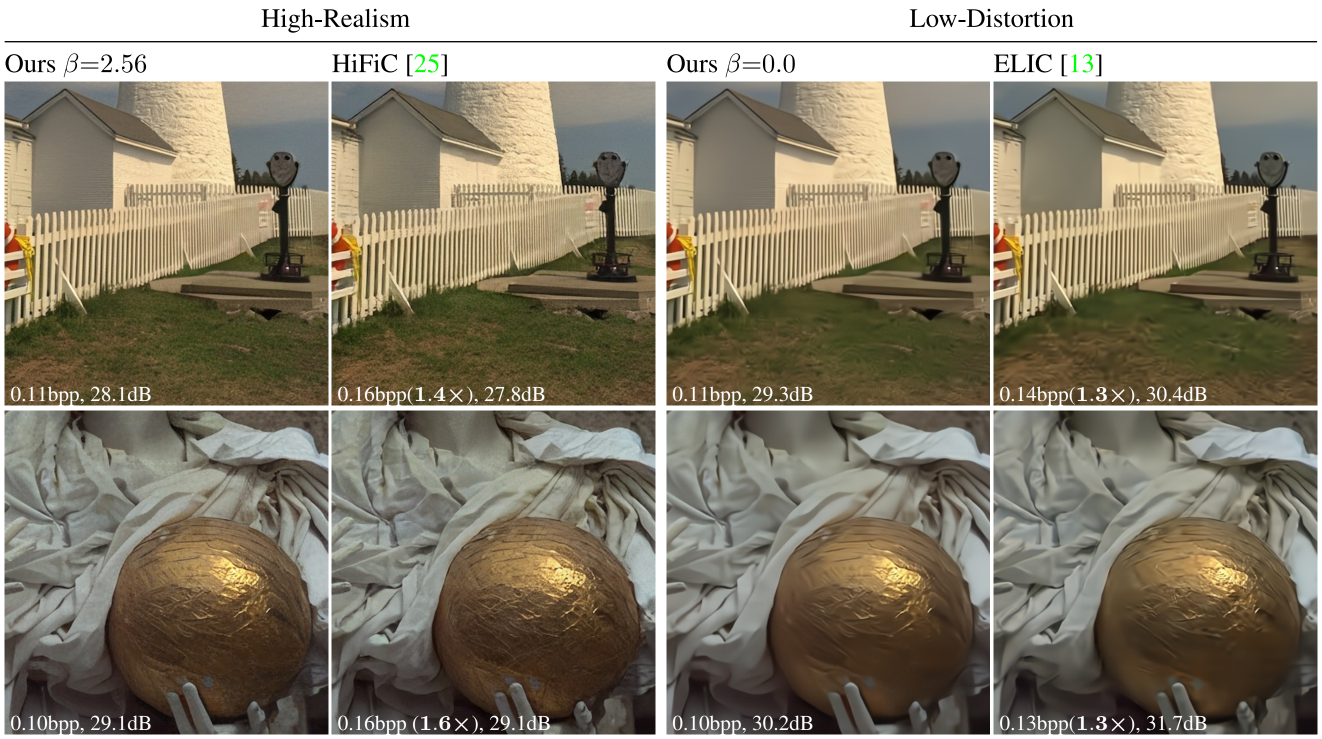
High-Realism
Low-Distortion
Ours β=2.56
HiFiC [25]
Ours β=0.0
ELIC [13]
0.11bpp, 28.1dB
0.16bpp($\mathbf{1.4}\times$), 27.8dB
0.11bpp, 29.3dB
0.14bpp($\mathbf{1.3}\times$), 30.4dB
0.10bpp, 29.1dB
0.16bpp ($\mathbf{1.6}\times$), 29.1dB
0.10bpp, 30.2dB
0.13bpp($\mathbf{1.3}\times$), 31.7dB

Figure 9. Comparing to the published HiFiC [25] and ELIC [13], both of which are state-of-the-art for realism and distortion, respectively. We show two reconstructions from our method, both obtained from the *same* representation, where we only vary the realism weight used by the generator. Note that PSNR increases as we move from a generative method to a purely MSE-optimized method, but our approach shows a higher PSNR than HiFiC even when using a high realism weight ($\beta = 2.56$) that leads to comparable perceptual quality. **This figure is best viewed zoomed in.**

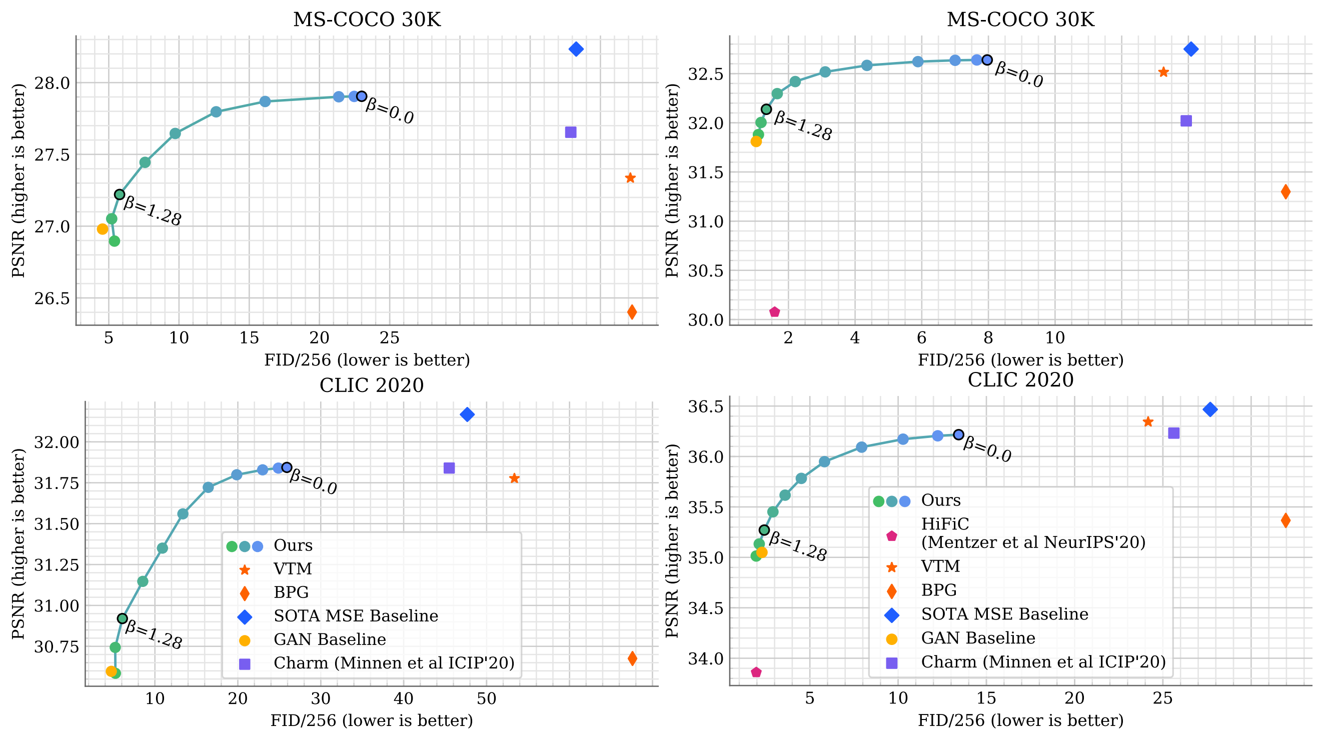

Figure 10. More rate points for distortion-realism, extending Fig. 4. The left column is at 0.17bpp for MS-COCO 30K, and 0.092bpp for CLIC 2020, whereas the right column is at 0.56bpp for MS-COCO 30K and 0.32bpp for CLIC 2020.

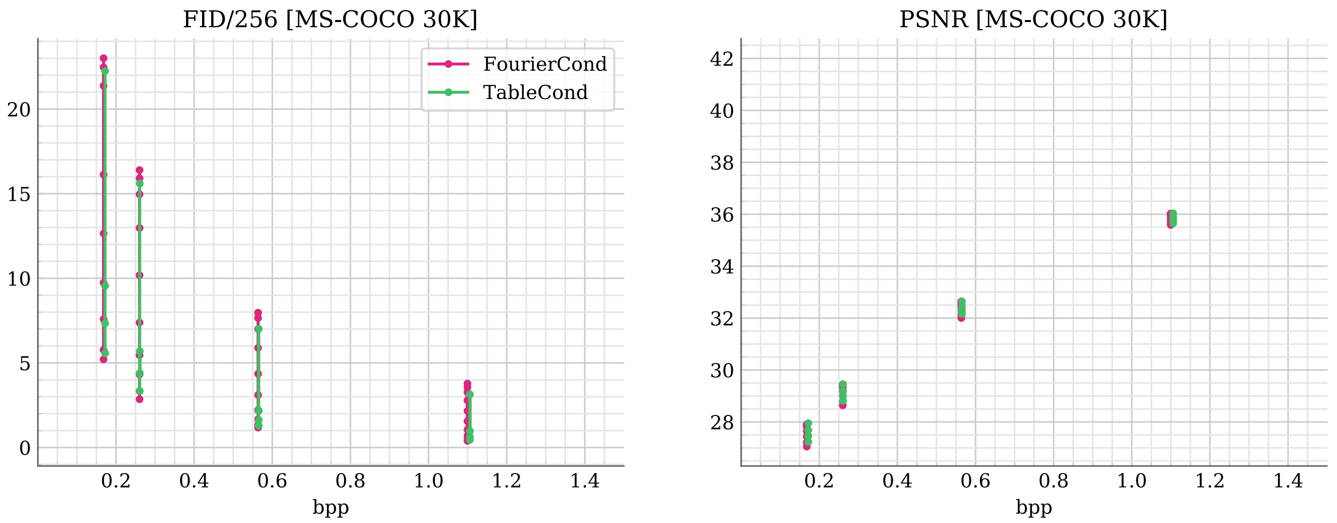

Figure 11. We compare the FourierCond and TableCond conditioning approaches for G. We find both give comparable results and choose FourierCond for our main model.