MSU Super-Resolution for Video Compression Benchmark 2021

G&M Lab head	Dr. Dmitriy Vatolin
Project adviser	Dr. Dmitriy Kulikov
Measurements & analysis	Evgeney Zimin, Egor Sklyarov

CS MSU Graphics & Media Lab, Video Group

31.08.2021

https://videoprocessing.ai/benchmarks/super-resolution-for-video-compression.html

sr-codecs-benchmark@videoprocessing.ai

Contents

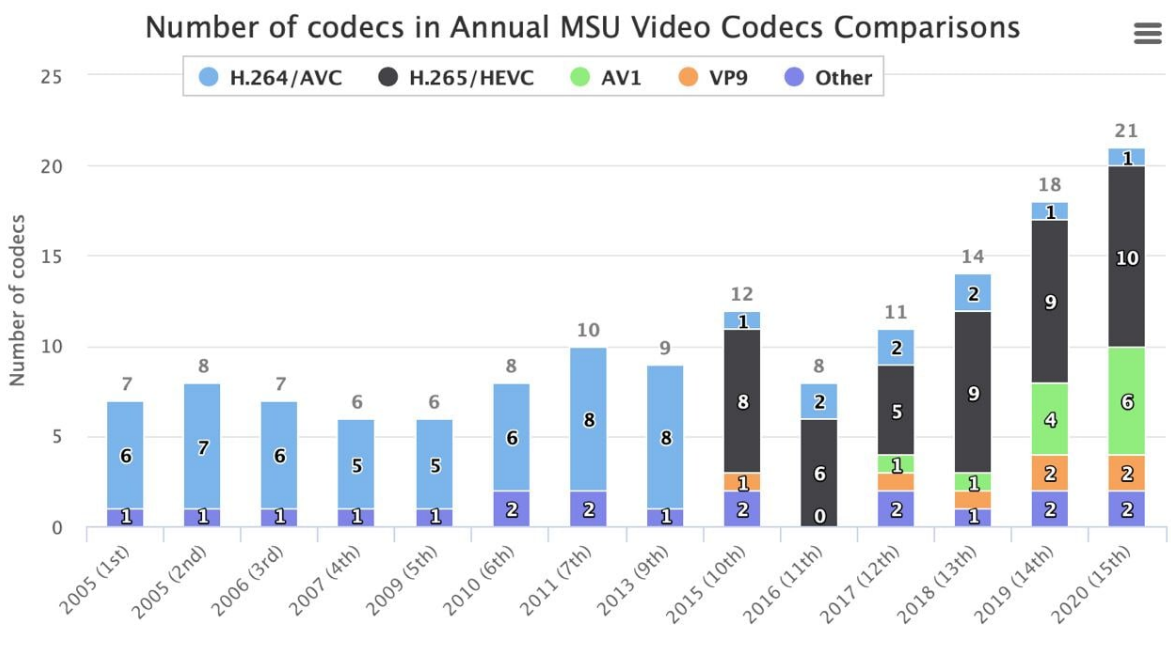
ABOUT GRAPHICS & MEDIA LAB VIDEO GROUP

Graphics & Media Lab Video Group is part of Lomonosov Moscow State University Computer Science Department. The history of MSU's Video Group began at the end of the 1980s, and Graphics & Media Lab was officially founded in 1998. The main research avenues of the lab include areas of computer graphics, computer vision, and video processing. Video Group works in areas of video compression (codecs testing and tuning, quality metrics research and development), video and S3D video quality analysis, and benchmarking.

Main research areas

Video Codecs Analysis

- Worldwide leading MSU Video Codecs Comparisons: conducted annually since 2007
- Collaborations with: **Google, Intel, AMD, NVIDIA, Huawei, Tencent, Alibaba,** and others
- The most detailed reports: 60+ Full HD and 4K videos, 8-10 different target bitrates, and 15000+ resulting graphs
- The most accurate subjective analysis powered by our Subjectify.us platform
- 27+ of our reports with more than 400.000 downloads are publically available
- **Codec tuning:** Our codec tuning improves encoding performance on a wide range of videos and encoding use cases, reducing bitrate up to 40% and encoding time up to 50% with the same quality



Video Quality Estimation

- MSU VQMT is a tool for performing **video/image quality analyses** using reference

or no-reference metrics
    - **Widest range of metrics and formats**, including 20+ objective metrics
    - **GPU support:** up to 11.7x speedup with GPU
    - Fastest implementation of VMAF
    - Fastest SSIM/MS-SSIM speed on 4K/8K video
- VQMT3D is a tool for stereoscopic video quality estimation
    - Detect **technical problems in stereoscopic movies** using the largest set of metrics
    - 14 metrics, including 5 unique ones, provide quality estimation of **2D-to-3D conversion**
    - Predict in advance whether you will have a headache after watching the 3D movie

Subjective Quality Estimation

- Subjectify.us is a **crowd-sourced evaluation platform** designed for subjective comparison of video, images, and sound processing methods
- **10x cut the budget** of your subjective study
- Previous cases of estimation:
    - Comparison of video completion methods
        - 6 competitors
        - 7 test-cases
        - 341 participant
    - Video codec comparison
        - 7 competitors (× 3 bitrates)
        - 4 test-cases
        - 325 participants (11530 answers)
    - Netflix study replication
        - 6-10 competitors (for various test-cases)
        - 7 test-cases
        - 375 participants

Other achievements

- **Cooperation with companies:** We have developed the exclusive tools for **Intel, Samsung, Huawei, RealNetworks**, and other companies, adapting our algorithms for specific video streams, applications, and hardware like TV sets, graphics cards, etc.
- **#1 Compression project:** our Compression project is the biggest resource on video compression
- **Benchmarking:** We launched 6 different benchmarks and expect to launch 14 more by the end of 2022. Our benchmarks include:
    - **Video Matting Benchmark:** The first public objective benchmark for video-matting methods
    - **Video Super-Resolution Benchmark**, which has the biggest amount of participants (20+ methods) on paperswithcode
    - **Deinterlacing Benchmark**: The first benchmark for comparison of deinterlacing methods

- **Video filters:** We make algorithms to correct and improve videos. Our filters have over 3 million downloads

90% of our projects are sponsored by companies. We have experience of a long-term collaboration with Intel, Samsung, Huawei, and others. All our research is aimed to be extremely practical for the industry. Our main sponsors and collaborators:

We are open to cooperation in the fields of video processing, video compression, and video quality analysis.

Feel free to contact us:

Main email:	video@compression.ru
Codec comparison:	videocodec-testing@graphics.cs.msu.ru
MSU benchmarks:	msu-benchmark@videoprocessing.ai
Personal email:	dmitriy@graphics.cs.msu.ru

1. OBJECTIVES AND TESTING RULES

1.1. Description

This report presents the results of **MSU Super-Resolution for Video Compression Benchmark**, in which we used objective metrics and subjective assessment to compare Super-Resolution (SR) models paired with different codecs.

Different SR models have different bitrate/quality tradeoffs when working with compressed video sequences. If two SRs produce results of the same subjective quality, the one that works with the lower bitrate input is considered to be better. Our benchmark aims to find the best SR+codec pair based on this criterion.

With the emergence of new video resolution standards, more efficient video encoding and decoding techniques are required. We can downscale a video before compression to lower the bitrate and then upscale the video to its original resolution using super-resolution methods. Our benchmark can help determine the best SR models to work with each of the different codec standards. This information will help make video coding with downsampling more effective.

The measurement process was as follows:

1. Source videos downscaling. We downscaled the videos to 960×540 resolution. You can see the information about the dataset in the overview section.
2. Compression of downscaled videos with different encoders. A list of codecs is presented in the overview section.
3. Upscaling with different super-resolution models. A list of models is presented in the overview section. Detailed description can be found in Appendix C.
4. Quality measurement of upscaled videos. We used full-reference metrics: Y-PSNR, YUV-MS-SSIM, Y-VMAF, Y-VMAF NEG, LPIPS, and ERQAv1.0 (see Appendix D). We also conducted a side-by-side subjective comparison for the videos by the subjectify.us platform.

Main points of comparison methodology:

- **Video dataset.** Our dataset is constantly being updated. Now it consists of 3×7×5=105 videos (3 Full HD GT videos in .yuv format decoded with 7 different bitrates using 5 different codecs). Videos were taken from MSU Video Codecs Comparison 2019 and 2020 test sets. The dataset contains videos in Full HD resolution with FPS from 24 to 30.
- **Low complexity of videos.** All videos have low SI/TI[1] value and simple textures. It was made to minimize compression artifacts that may occur to make restoration of details possible.
- **Upscale factor**. We are currently testing only 2x upscale.

[1] Y. Wang, S. Inguva, B. Adsumilli, “YouTube UGC dataset for video compression research,” in *IEEE 21st International Workshop on Multimedia Signal Processing*, 2019

- **Ranking**: As an overall score indication, an approach we called BSQ-rate[2] (bitrate-for-the-same-quality rate) was used.

1.2. Video Sequences

Brief descriptions of the video sequences used in our comparison appear in Table 1. Appendix A provides more detailed descriptions of these sequences.

	Sequence	Number of frames	Framerate	Resolution
1.	animation_clip	100	30	1920×1080
2.	skiing	179	24	1920×1080
3.	street_show	200	24	1920×1080

Table 1: Summary of video sequences

1.3. Codecs

Brief descriptions of codecs used in our comparison appear in Table 2. Appendix B provides detailed descriptions of all codecs in our comparison.

Codec	Standard	Implementation
x264	H.264	FFmpeg version 4.2.4
x265	H.265	FFmpeg version 4.2.4
aomenc	AV1	FFmpeg version 4.2.4
VVenC	H.266	https://github.com/fraunhoferhhi/vvenc v1.0.0
uavs3e	AVS3	https://github.com/uavs3/uavs3e cd29508

Table 2: Short codecs' description

[2] A. V. Zvezdakova, D. L. Kulikov, S. V. Zvezdakov, D. S. Vatolin, "BSQ-rate: a new approach for video-codec performance comparison and drawbacks of current solutions," in *Programming and computer software*, 2020

1.4. Participants

Brief descriptions of the Super-Resolution models used in our comparison appear in Table 3. Appendix C provides detailed descriptions of all models in our comparison.

Model	**VSR or SISR**	**Upscale factor**	**Year**
ahq-11	VSR	2x, 4x	2021
amq-12	VSR	2x, 4x	2021
amqs-1	VSR	2x, 4x	2021
DBVSR	VSR	4x	2020
DynaVSR-R	VSR	4x	2020
EGVSR	VSR	4x	2021
iSeeBetter	VSR	4x	2020
LGFN	VSR	4x	2020
RealSR	SISR	4x	2020
SOF-VSR-BD	VSR	4x	2020
SOF-VSR-BI	VSR	4x	2020
waifu2x-anime	SISR	2x, 4x	2018
waifu2x-cunet	SISR	2x, 4x	2018

Table 3: Description of participants' Super-Resolution models

2. OBJECTIVE COMPARISON

2.1. Methodology

Firstly, we downscale our Full HD GT video using FFmpeg to make it 960×540 resolution. Then, we compress scaled video with seven different bitrates (approximately 100, 300, 600, 1000, 2000, 4000, and 6000 kbps). The resulting videos are given as input to a Super-Resolution model (See Figure 1).

In our benchmark we test 2x upscale, however, there are some Super-Resolution models which can only do 4x upscale. In this case, we downscale these models' results to Full HD by using FFmpeg.

We also compress Full HD GT video without scaling to make "only compressed" results (see Figure 2).

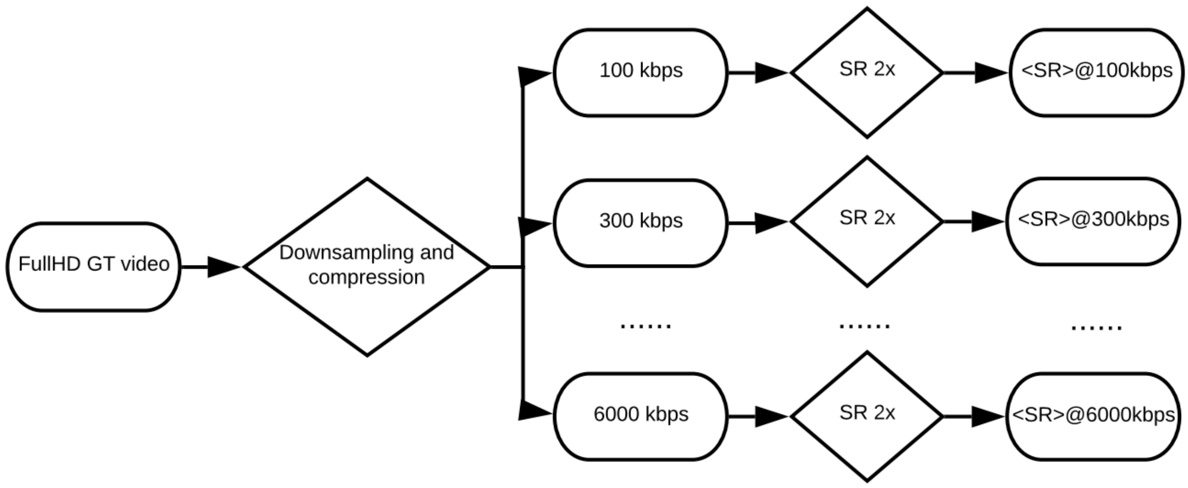


Figure 1: SR results evaluation steps

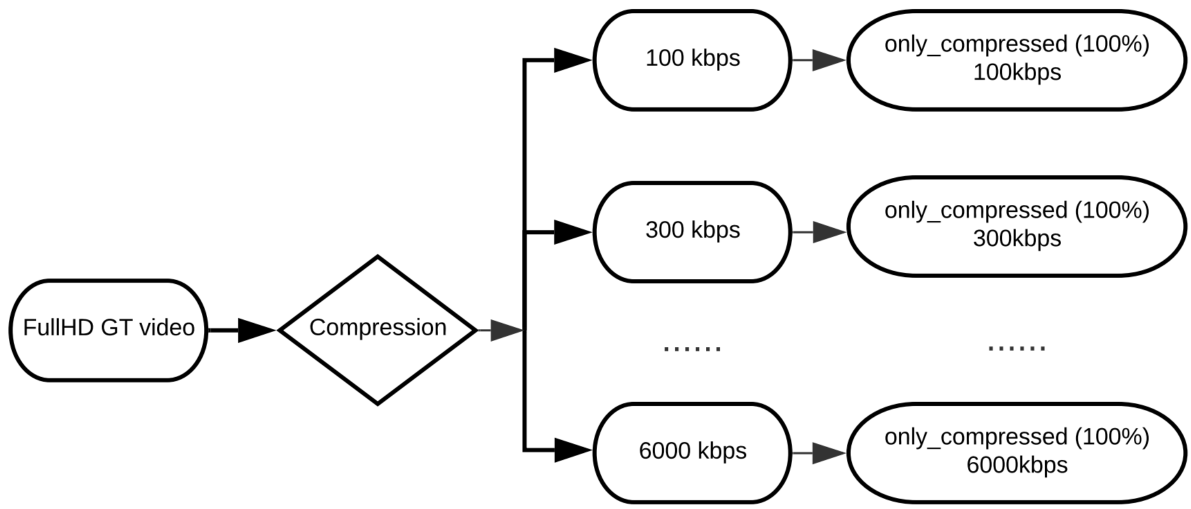


Figure 2: "only compressed" evaluation steps

Next, we calculate each metric for each result (including “only compressed”). We calculate shifted Y-PSNR, shifted YUV-MS-SSIM, Y-VMAF, Y-VMAF NEG, LPIPS, and ERQAv1.0 (see Appendix D). Then, we build RD (Rate-Distortion) curves and calculate BSQ-rate[3] for each metric. We take the “only compressed” result as a reference during the calculations.

2.2. x264 results

In this section, you can see the results of applying SR models on videos compressed with the x264 codec. The explanation of measuring is presented in Section E.2.

The RD charts show variation in SR models’ quality by bitrate. For each metric, a higher value presumably indicates better quality.

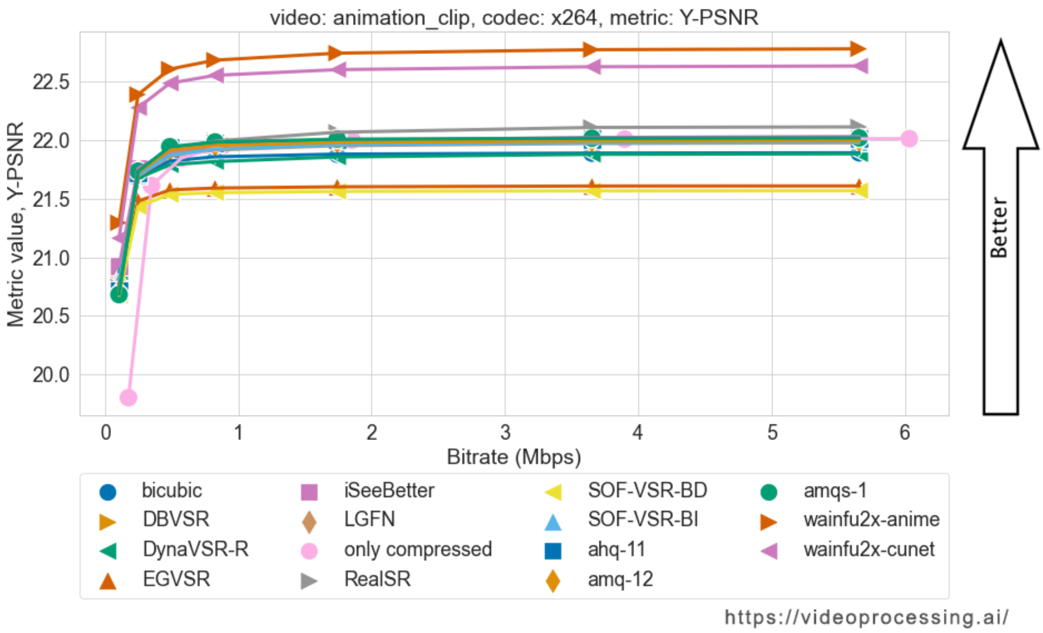


Figure 3a: Bitrate/Quality — *animation_clip* sequence, x264 codec, Y-PSNR metric

[3] A. V. Zvezdakova, D. L. Kulikov, S. V. Zvezdakov, D. S. Vatolin, "BSQ-rate: a new approach for video-codec performance comparison and drawbacks of current solutions," in *Programming and computer software*, 2020

Barcharts show the average BSQ-rate of each model relative to “only compressed”. A lower BSQ-rate presumably indicates a better bitrate/quality tradeoff. BSQ-rate can be equal to infinity if the model's RD curve does not intersect with the reference curve.

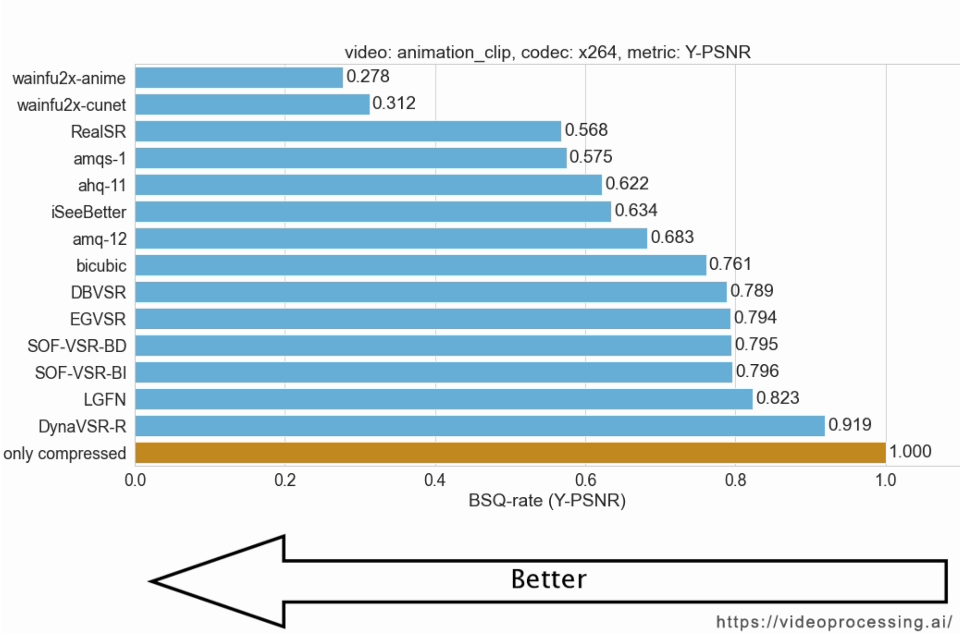


Figure 3b: BSQ-rate relative to “only compressed” — *animation_clip* sequence, x264 codec, Y-PSNR metric

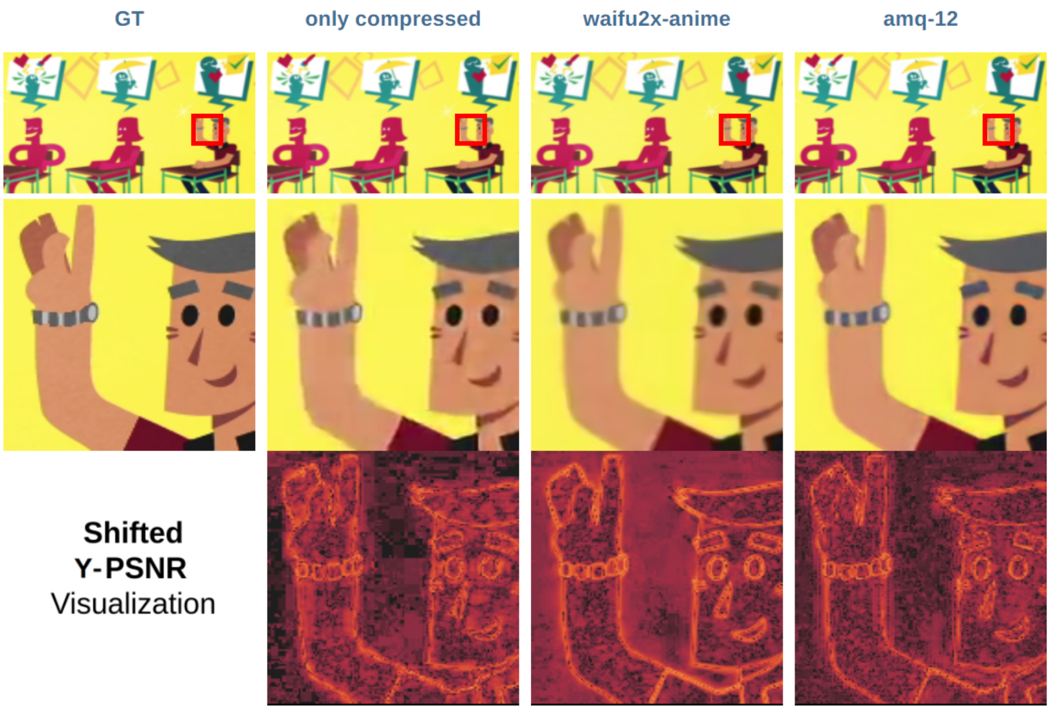


Figure 3c: Visual comparison between a few methods and shifted Y-PSNR visualization — *animation_clip* sequence, x264 codec

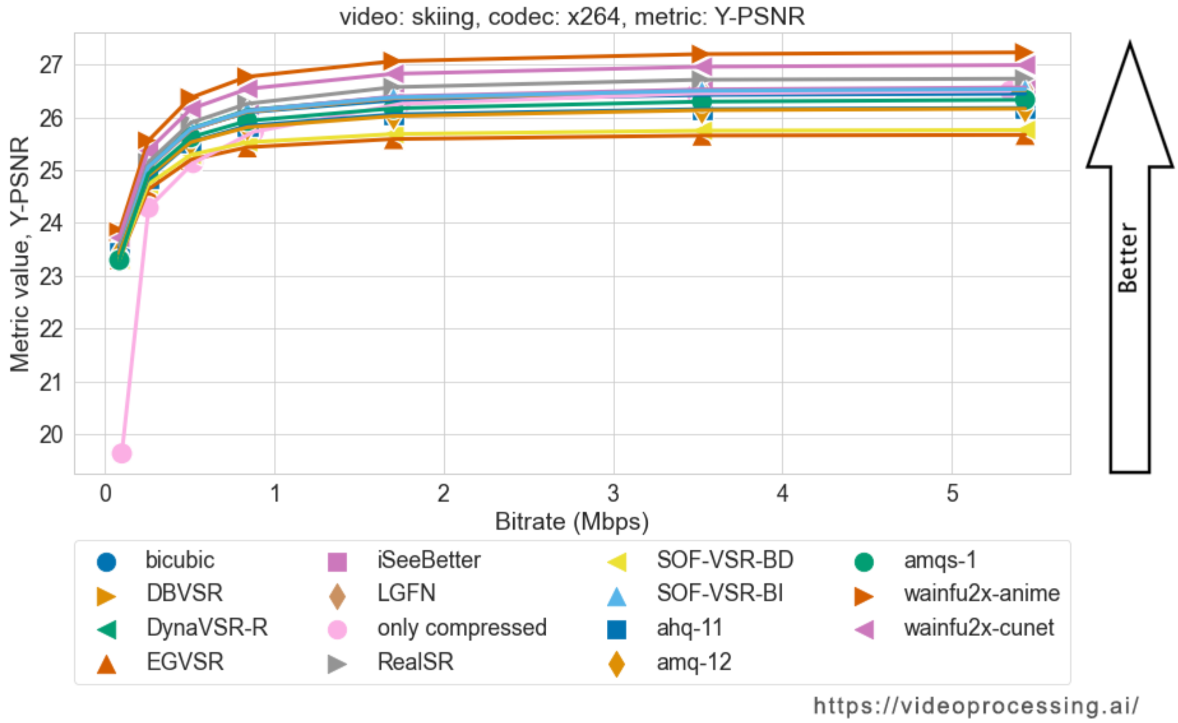


Figure 4a: Bitrate/Quality — *skiing* sequence, x264 codec, Y-PSNR metric

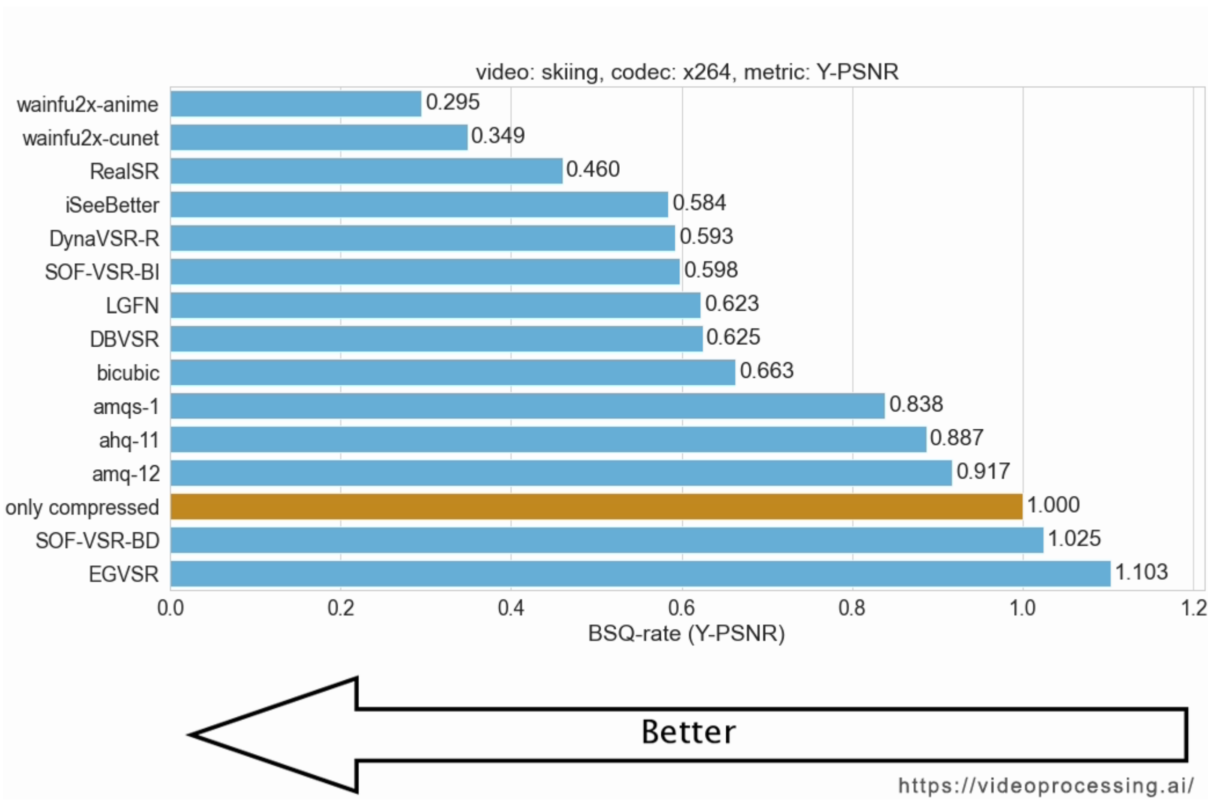


Figure 4b: BSQ-rate relative to “only compressed” —
skiing sequence, x264 codec, Y-PSNR metric

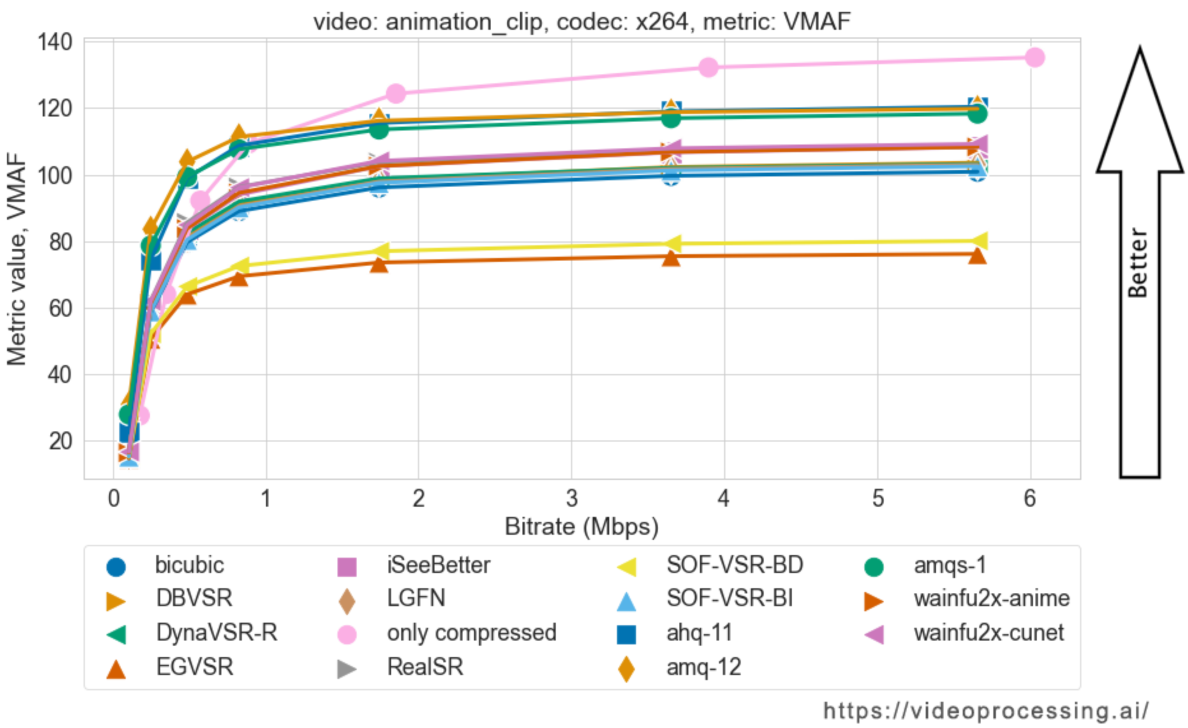


Figure 5a: Bitrate/Quality — *animation_clip* sequence, x264 codec, VMAF metric

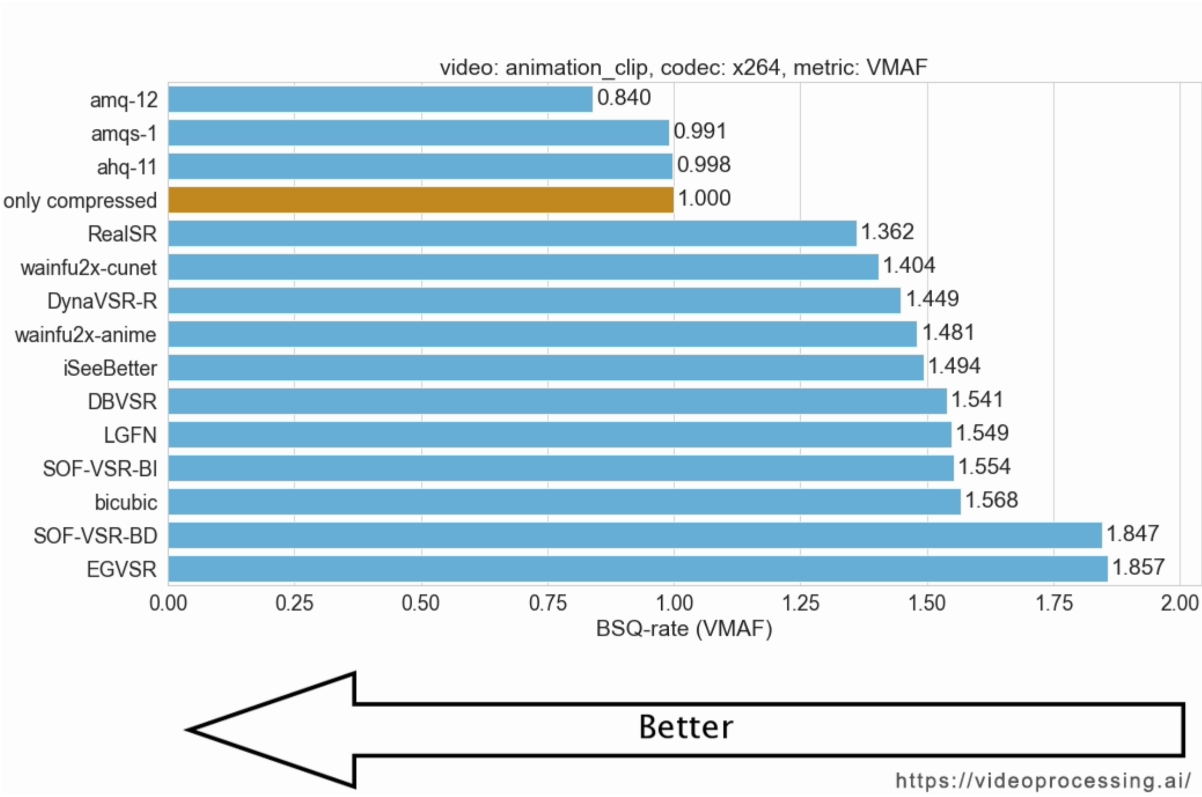


Figure 5b: BSQ-rate relative to “only compressed” — *animation_clip* sequence, x264 codec, VMAF metric

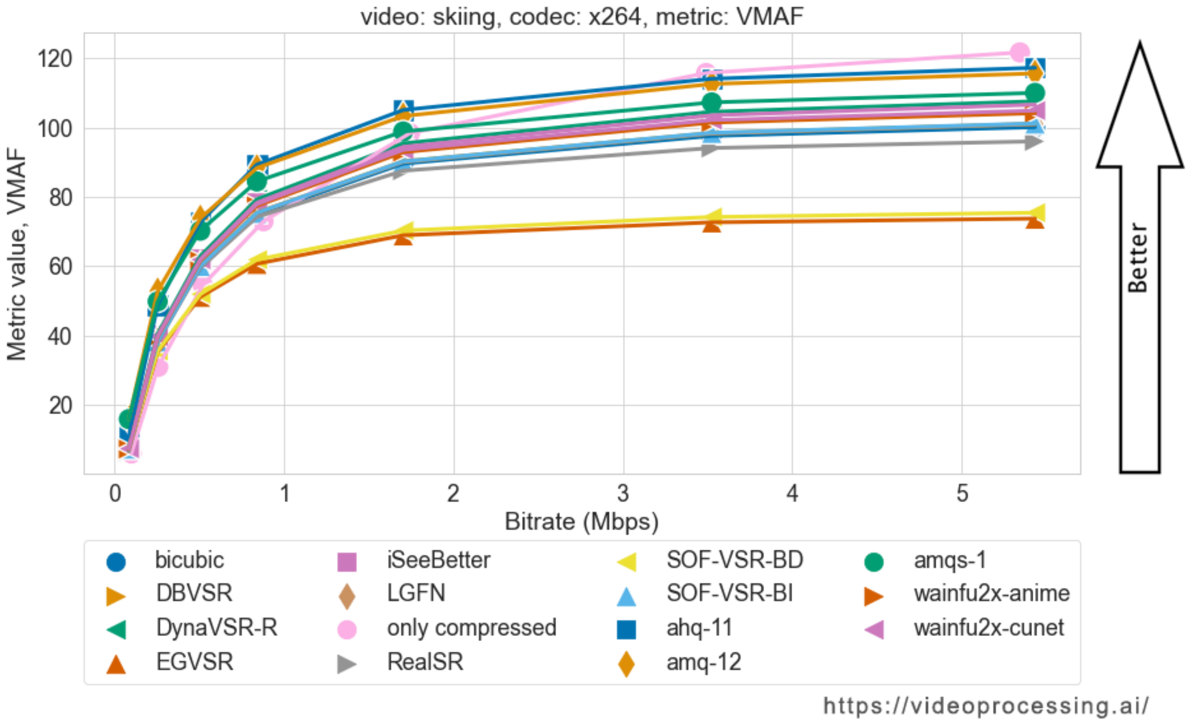


Figure 6a: Bitrate/Quality — *skiing* sequence, x264 codec, VMAF metric

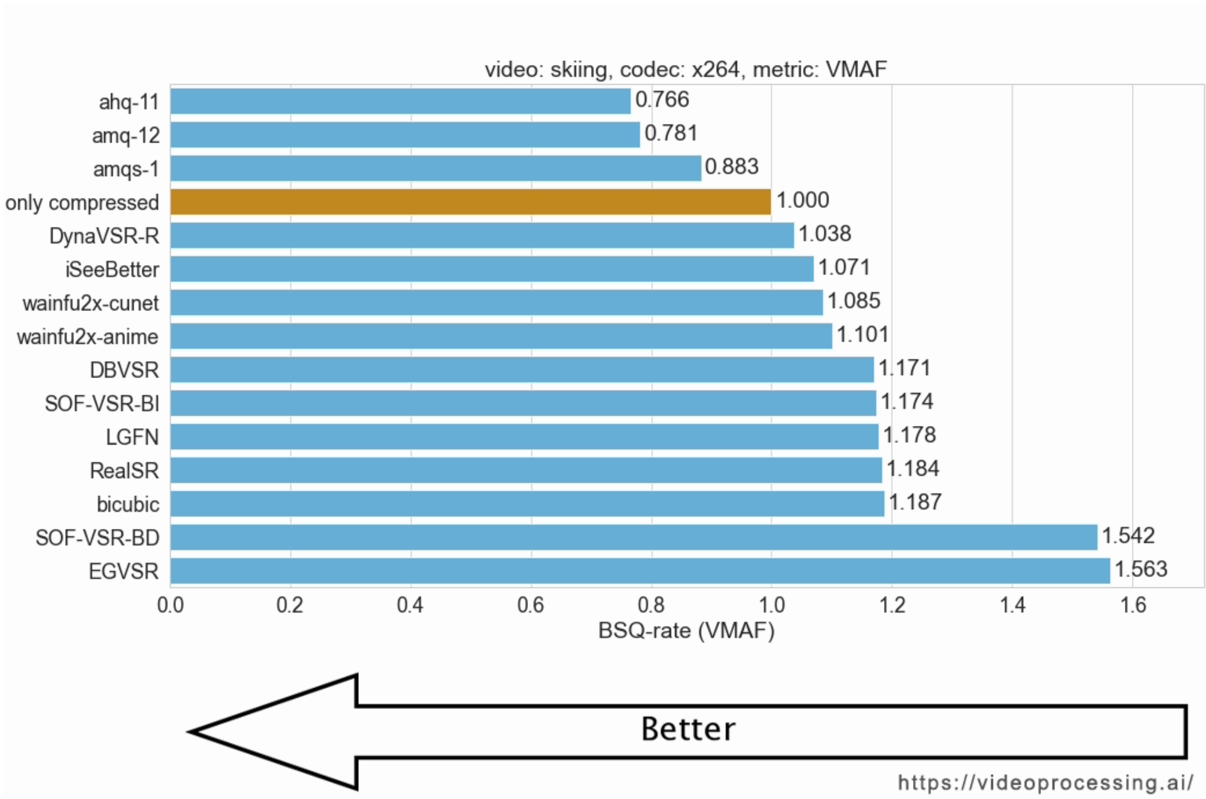


Figure 6b: BSQ-rate relative to "only compressed" —
skiing sequence, x264 codec, VMAF metric

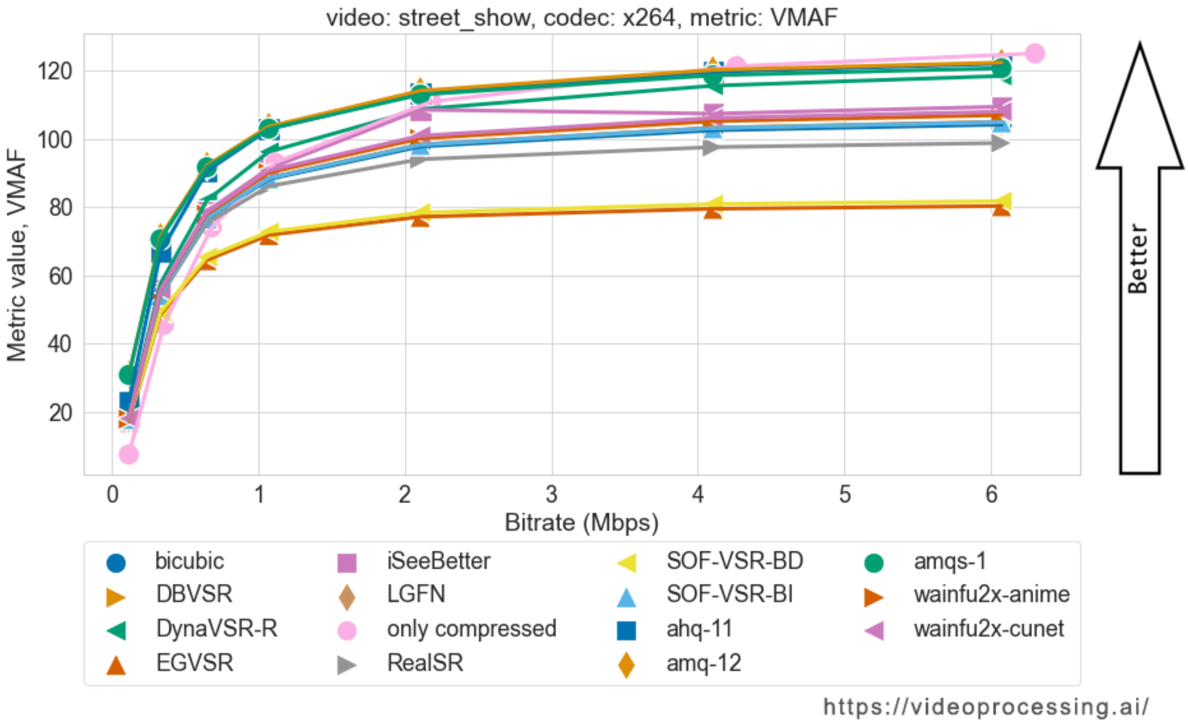


Figure 7a: Bitrate/Quality — *street_show* sequence, x264 codec, VMAF metric

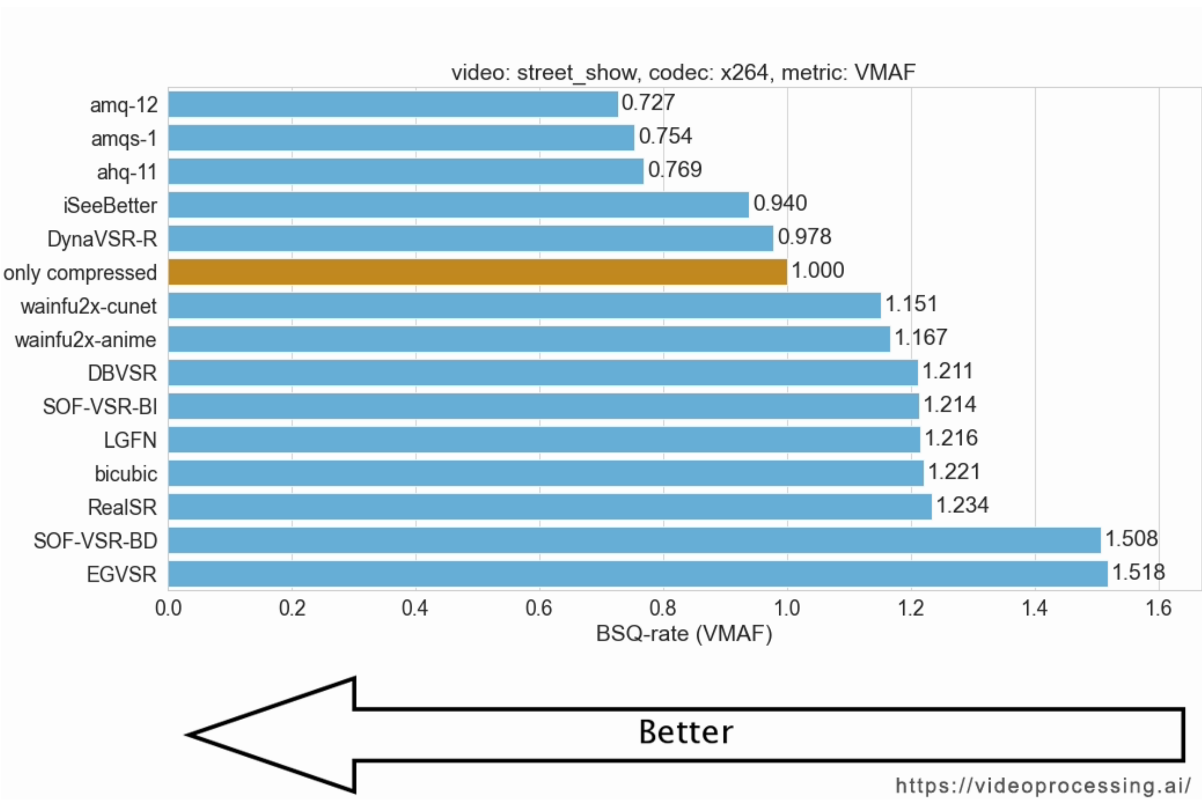


Figure 7b: BSQ-rate relative to “only compressed” — *street_show* sequence, x264 codec, VMAF metric

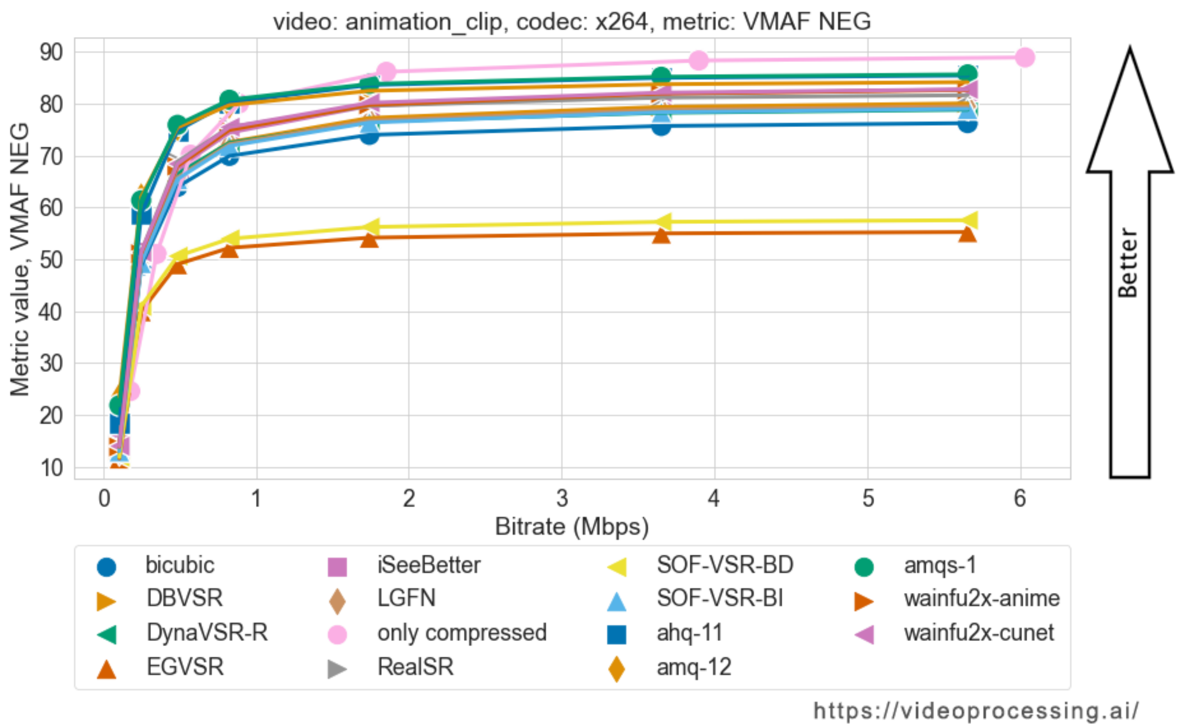


Figure 8a: Bitrate/Quality — *animation_clip* sequence, x264 codec, VMAF NEG metric

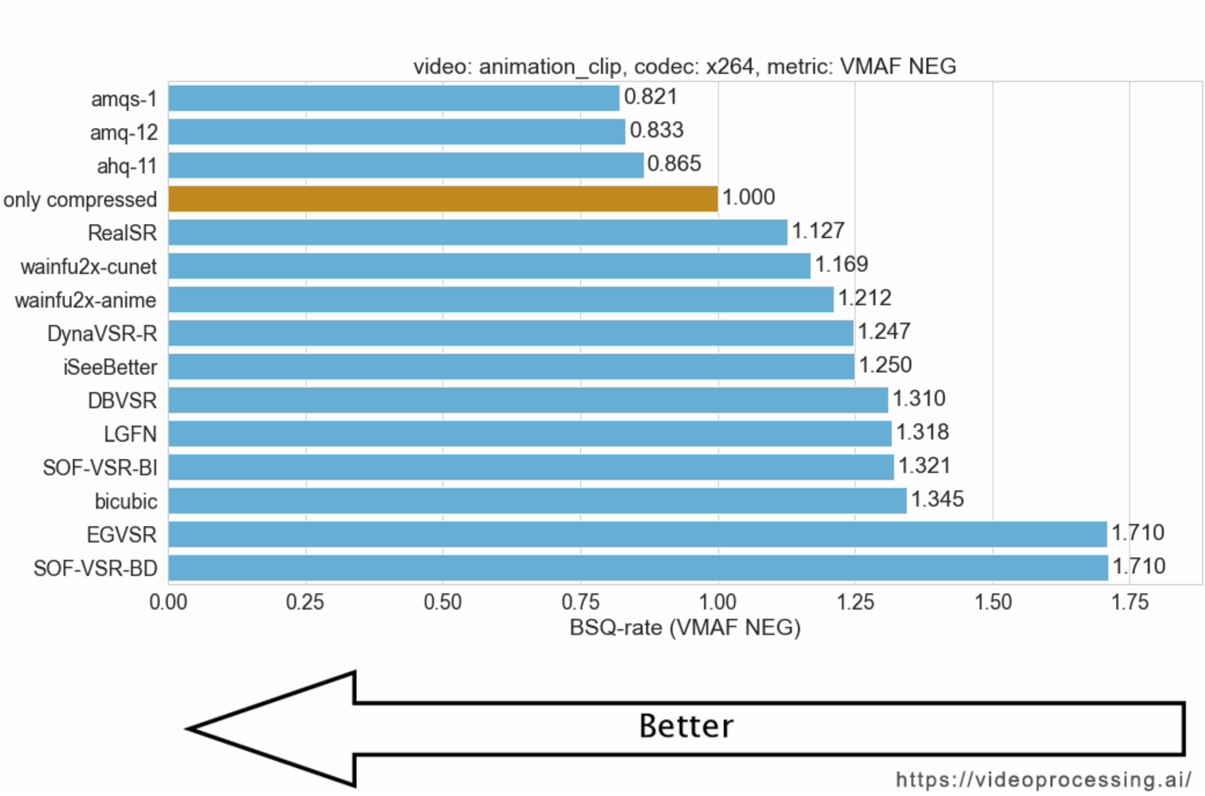


Figure 8b: BSQ-rate relative to “only compressed” —
animation_clip sequence, x264 codec, VMAF NEG metric

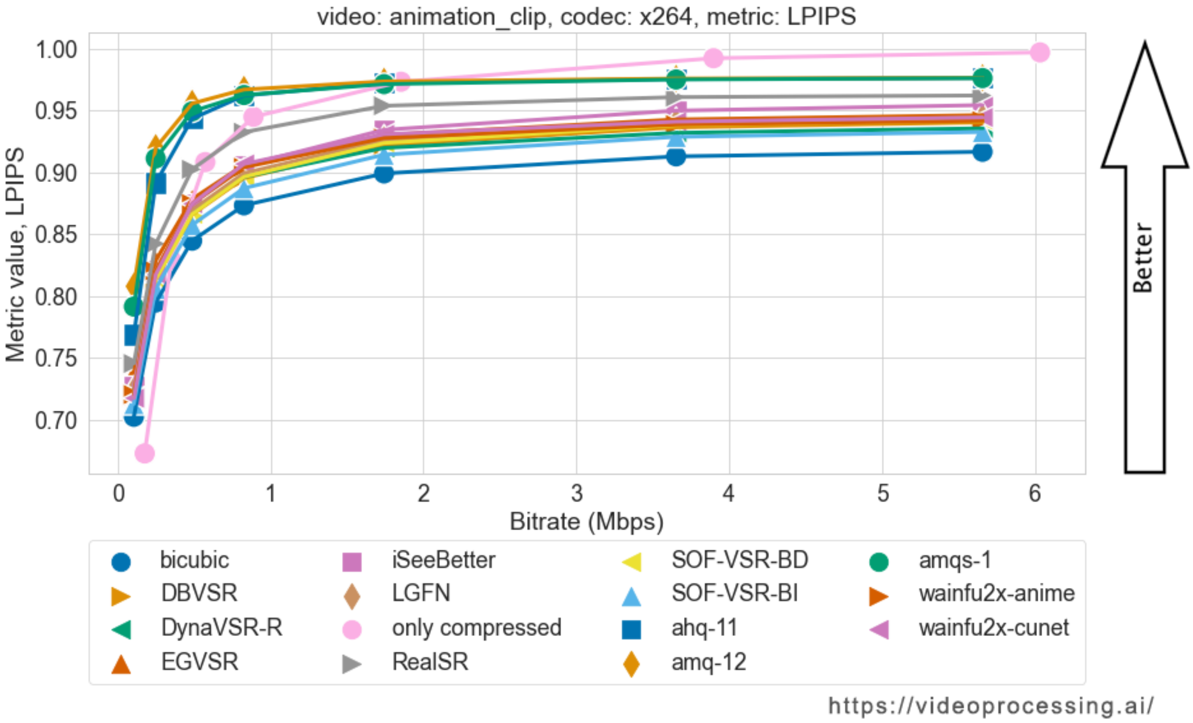


Figure 9a: Bitrate/Quality — *animation_clip* sequence, x264 codec, LPIPS metric

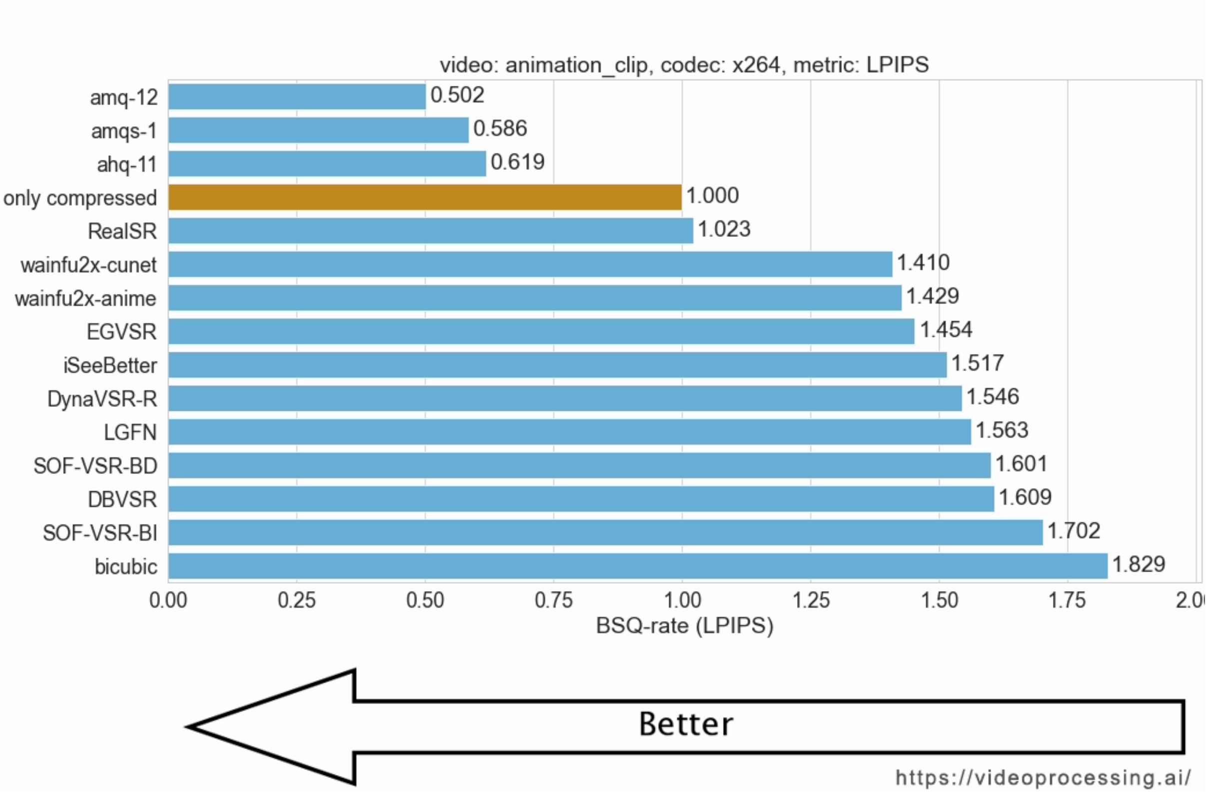


Figure 9b: BSQ-rate relative to "only compressed" — *animation_clip* sequence, x264 codec, LPIPS metric

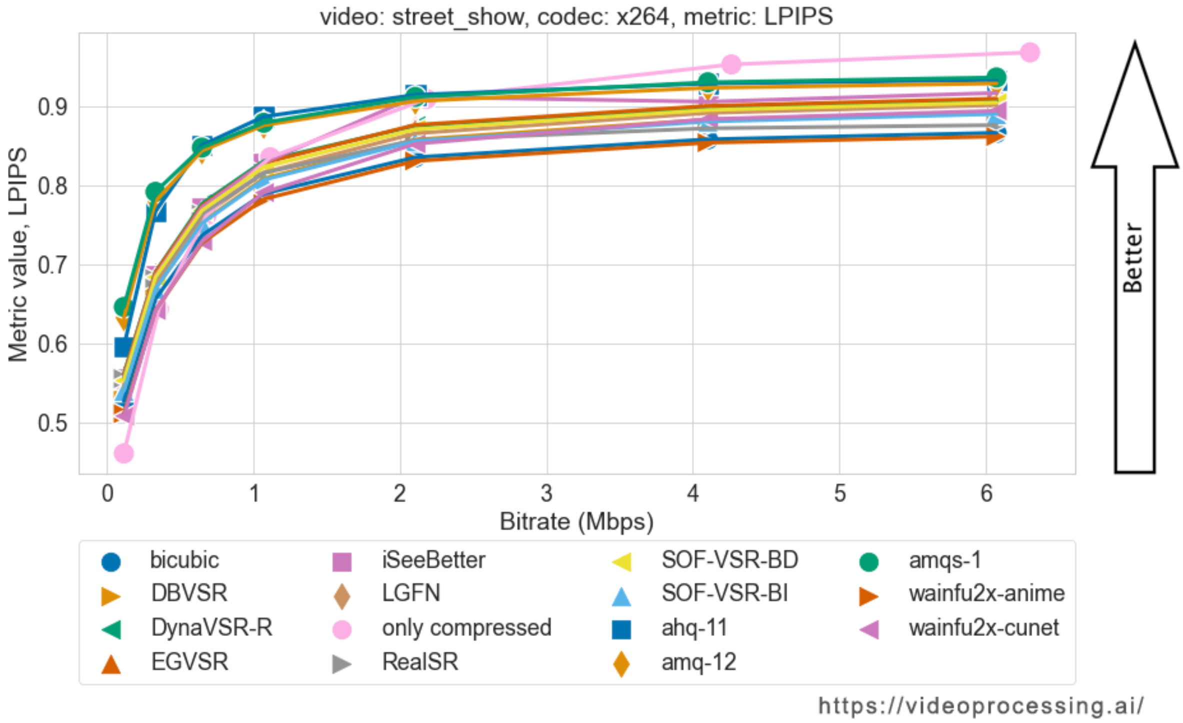


Figure 10a: Bitrate/Quality — *street_show* sequence, x264 codec, LPIPS metric

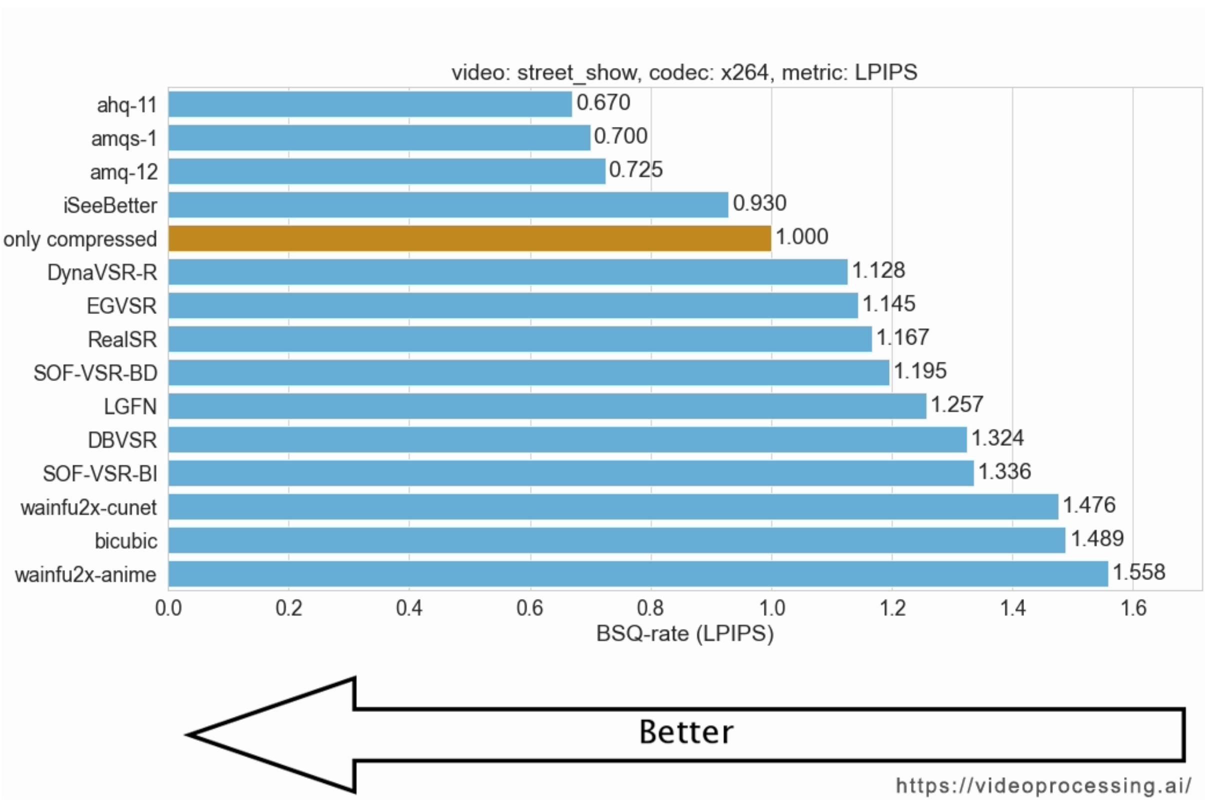


Figure 10b: BSQ-rate relative to “only compressed” — *street_show* sequence, x264 codec, LPIPS metric

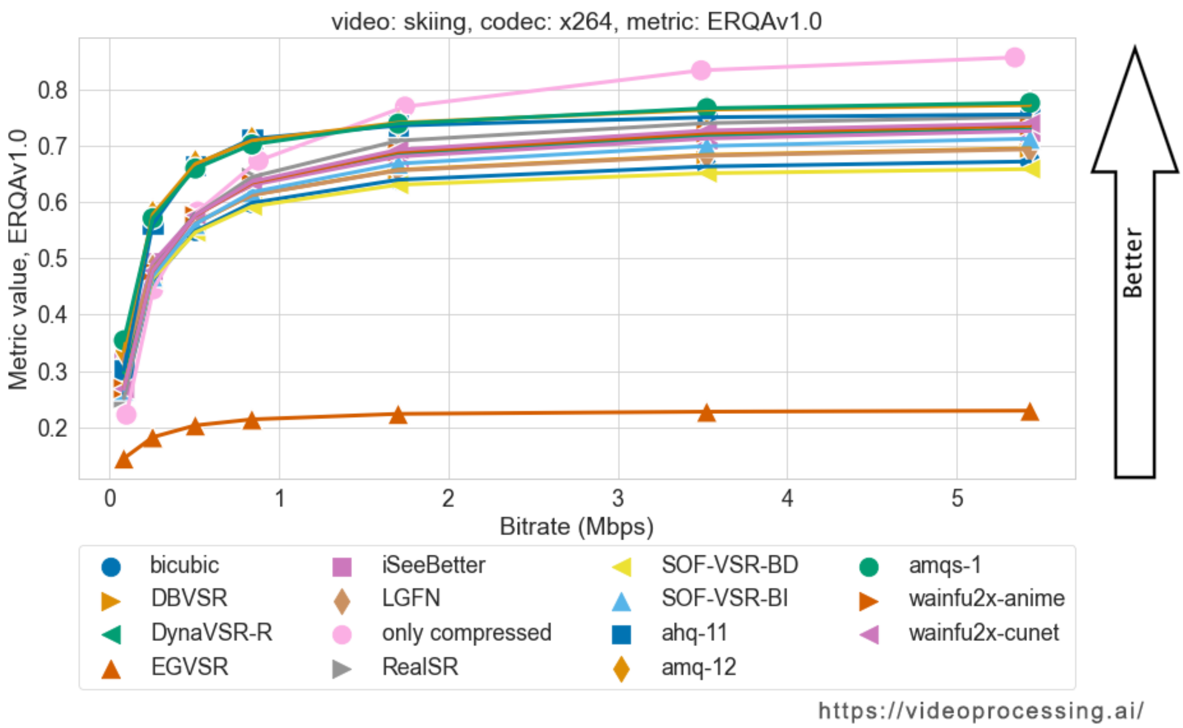


Figure 11a: Bitrate/Quality — *skiing* sequence, x264 codec, ERQAv1.0 metric

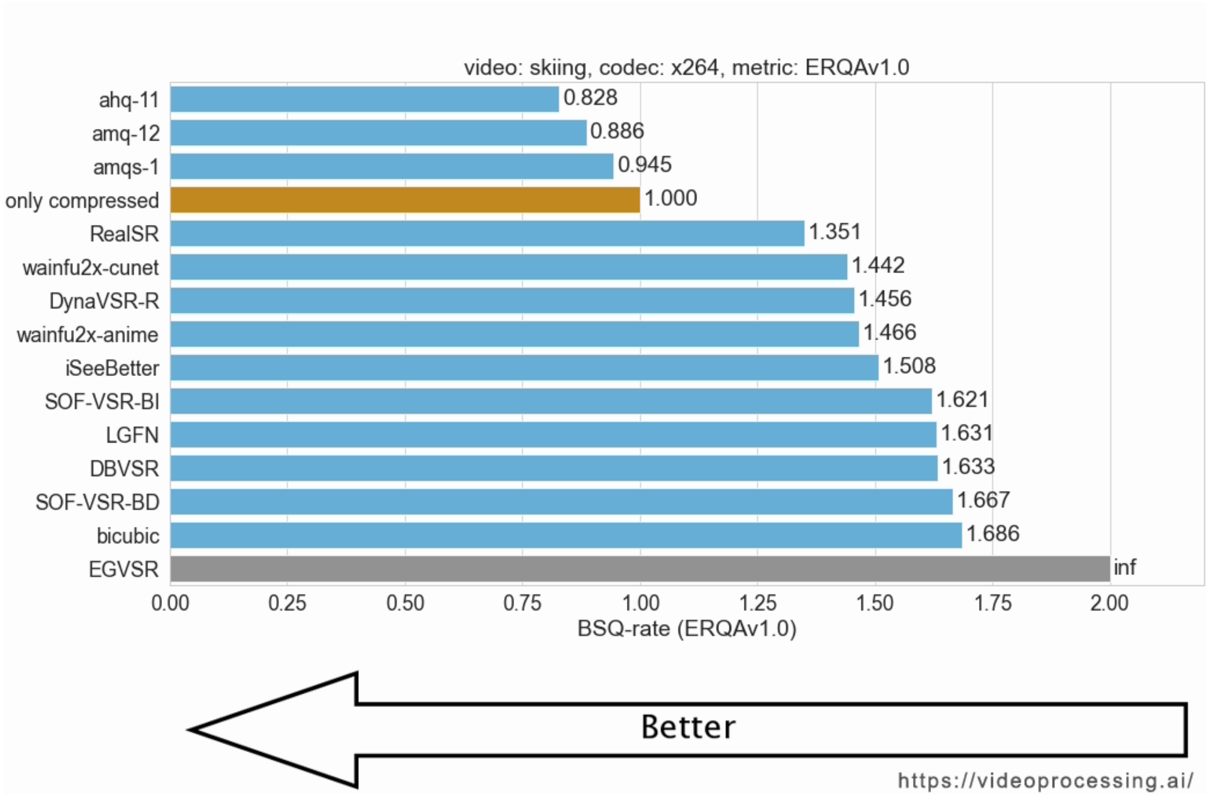


Figure 11b: BSQ-rate relative to “only compressed” —
skiing sequence, x264 codec, ERQAv1.0 metric

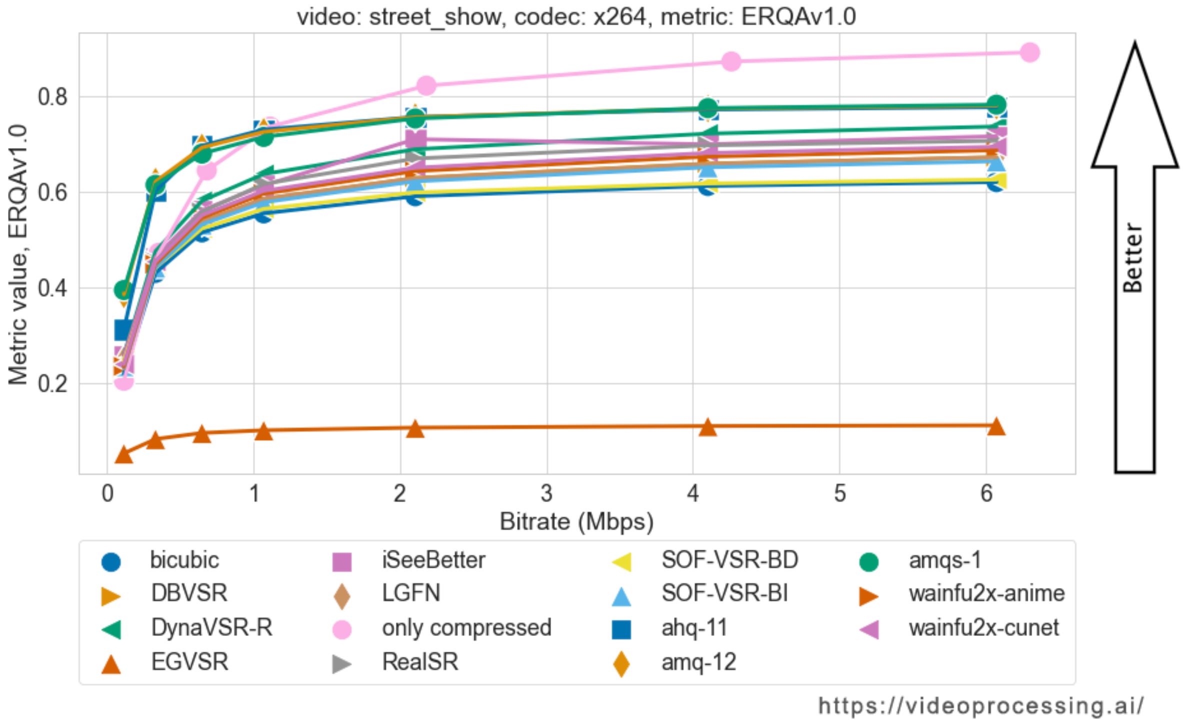


Figure 12a: Bitrate/Quality — *street_show* sequence, x264 codec, ERQAv1.0 metric

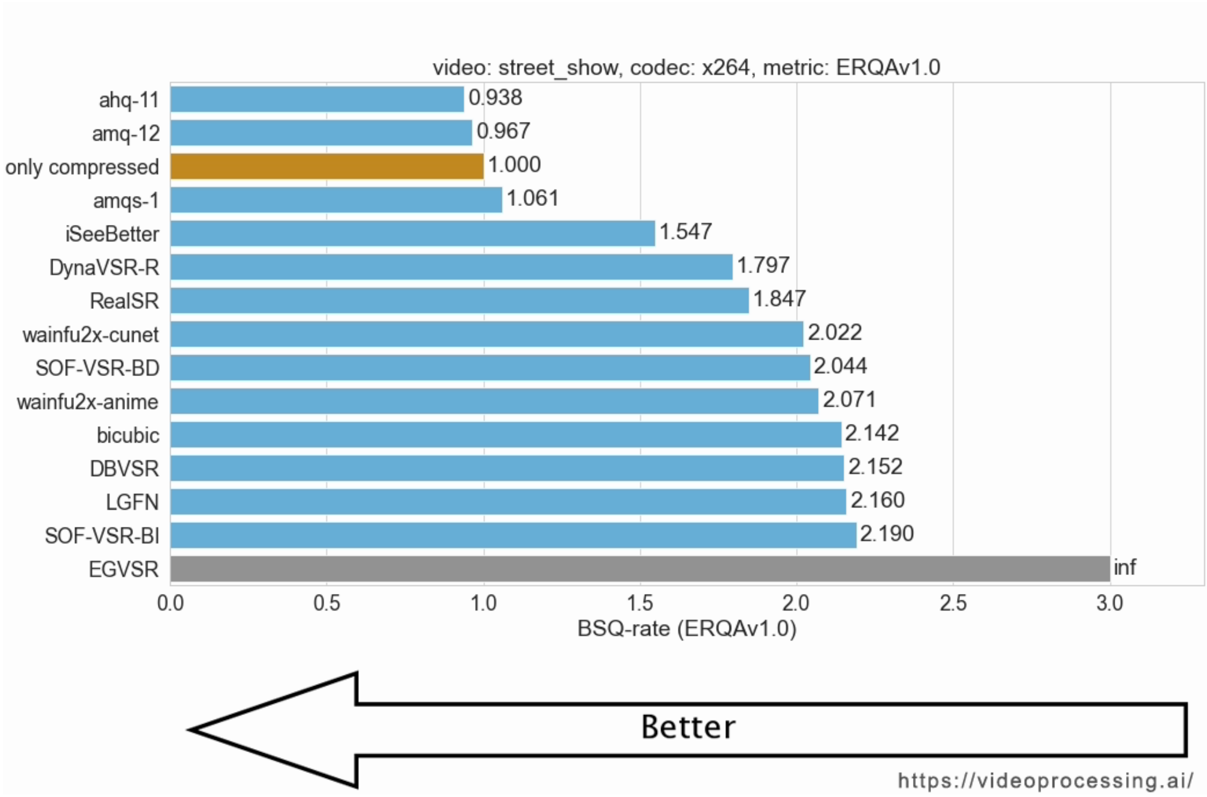


Figure 12b: BSQ-rate relative to “only compressed” —
street_show sequence, x264 codec, ERQAv1.0 metric

In Figure 13 you can see the average BSQ-rate over each metric for the x264 codec. “Only compressed” made by the x264 codec was used as a reference.

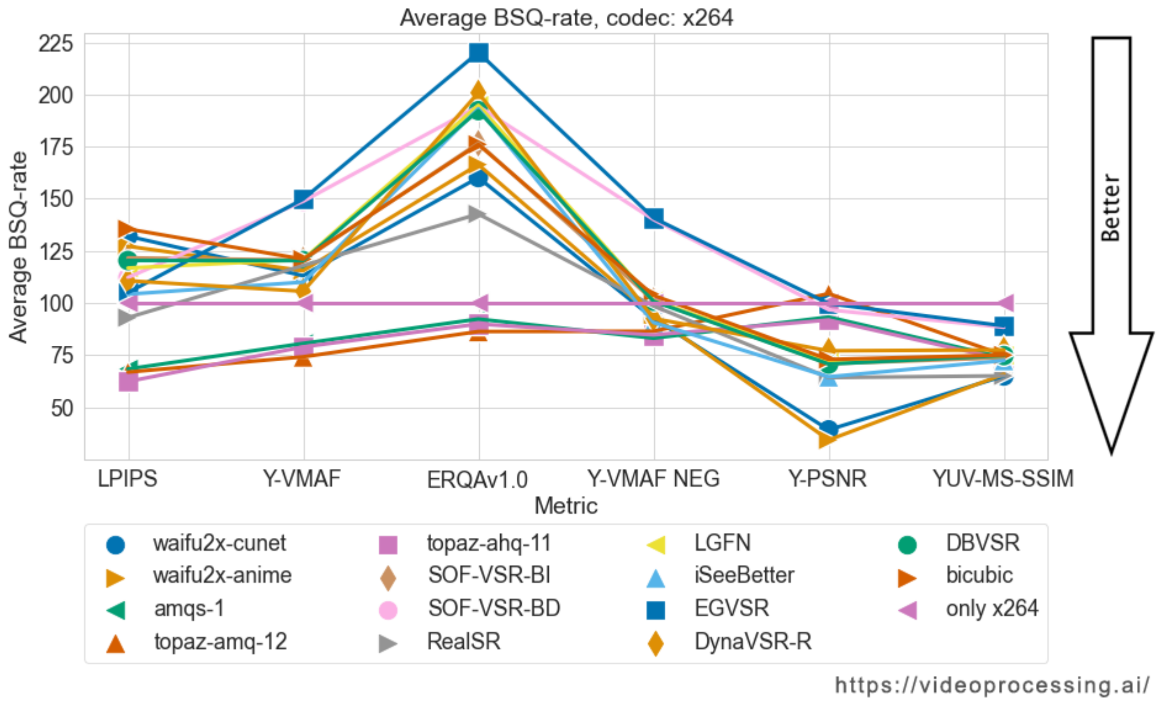


Figure 13: Average BSQ-rate relative to “only x264”.
SR input was compressed with the x264 codec

2.3. x265 results

In this section, you can see the results of applying SR models on videos compressed with the x265 codec.

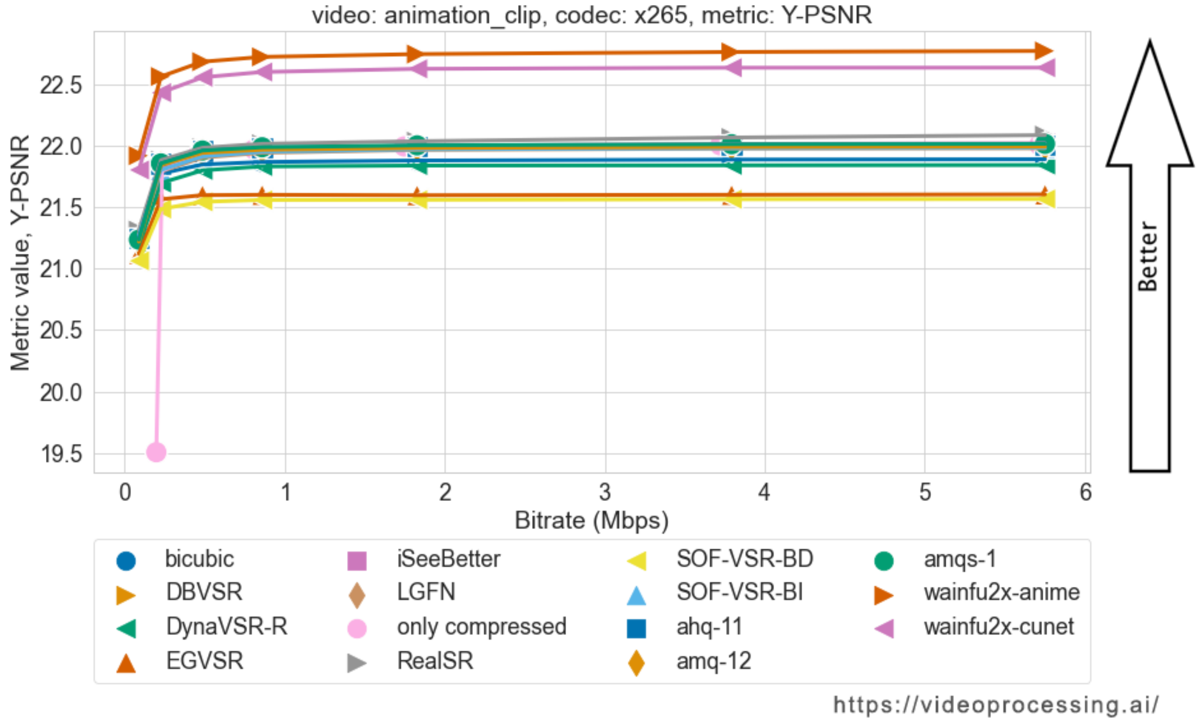


Figure 14a: Bitrate/Quality — *animation_clip* sequence, x265 codec, Y-PSNR metric

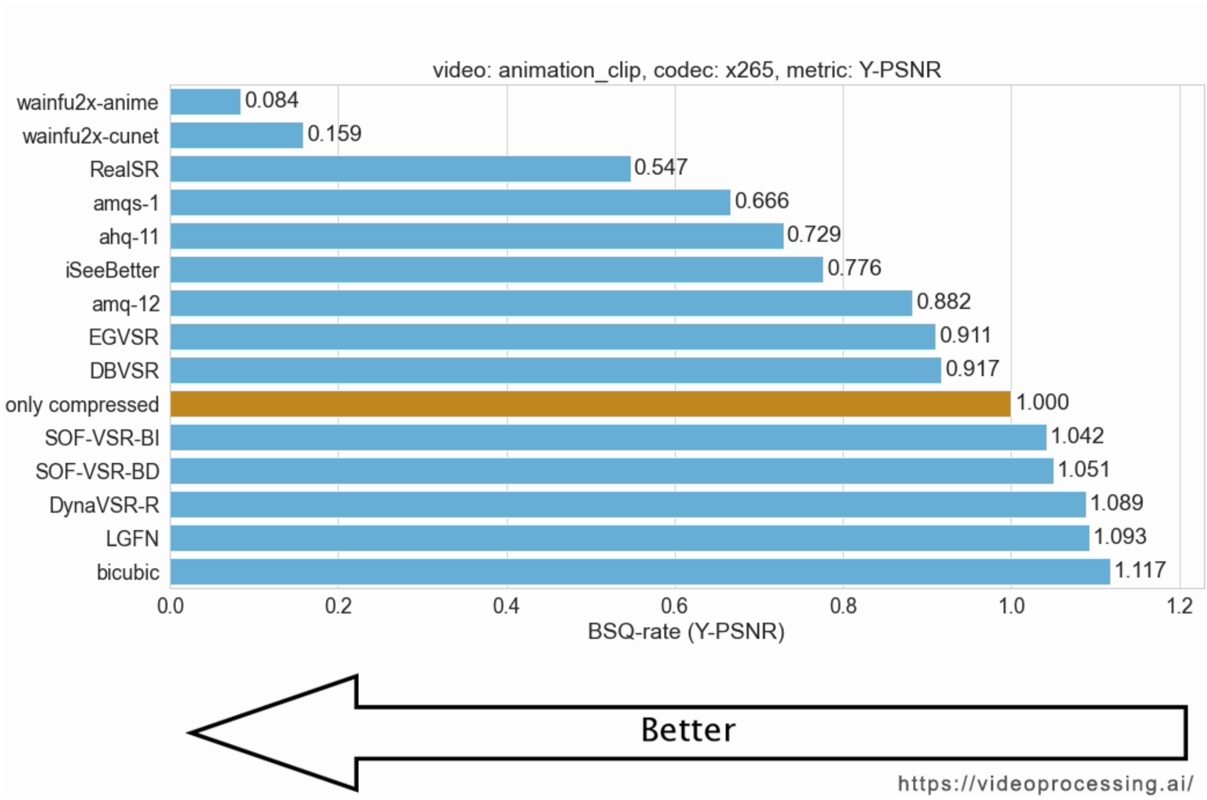


Figure 14b: BSQ-rate relative to “only compressed” — *animation_clip* sequence, x265 codec, Y-PSNR metric

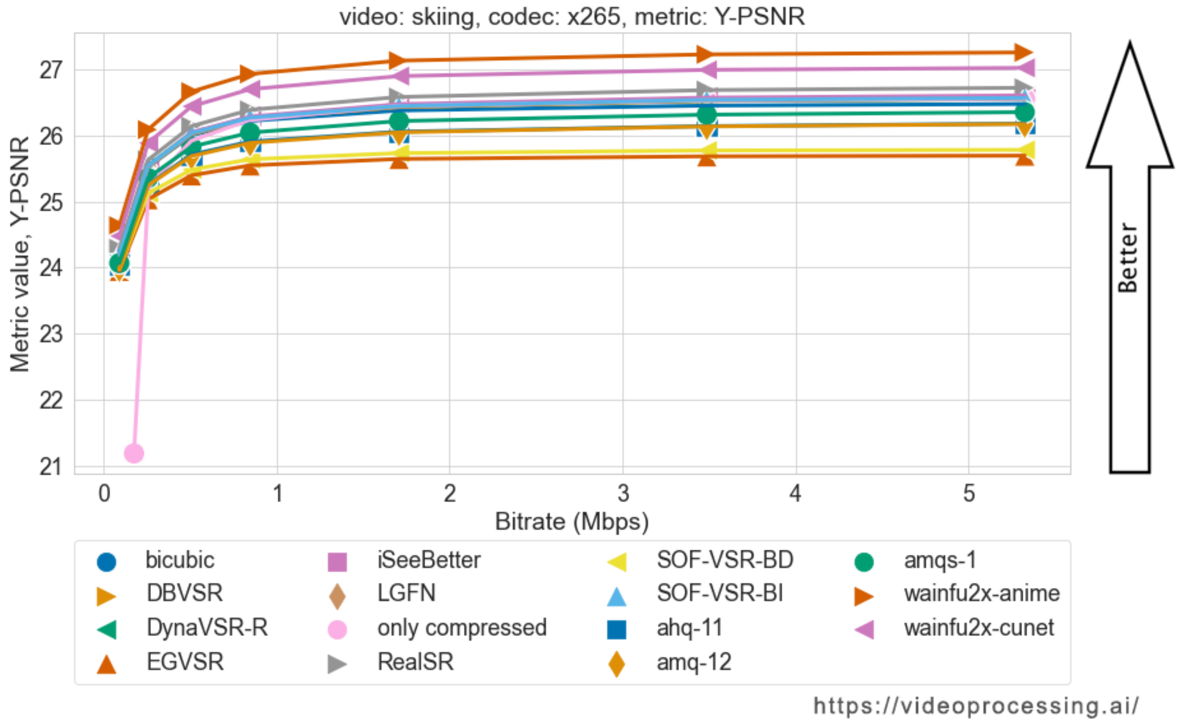


Figure 15a: Bitrate/Quality — *skiing* sequence, x265 codec, Y-PSNR metric

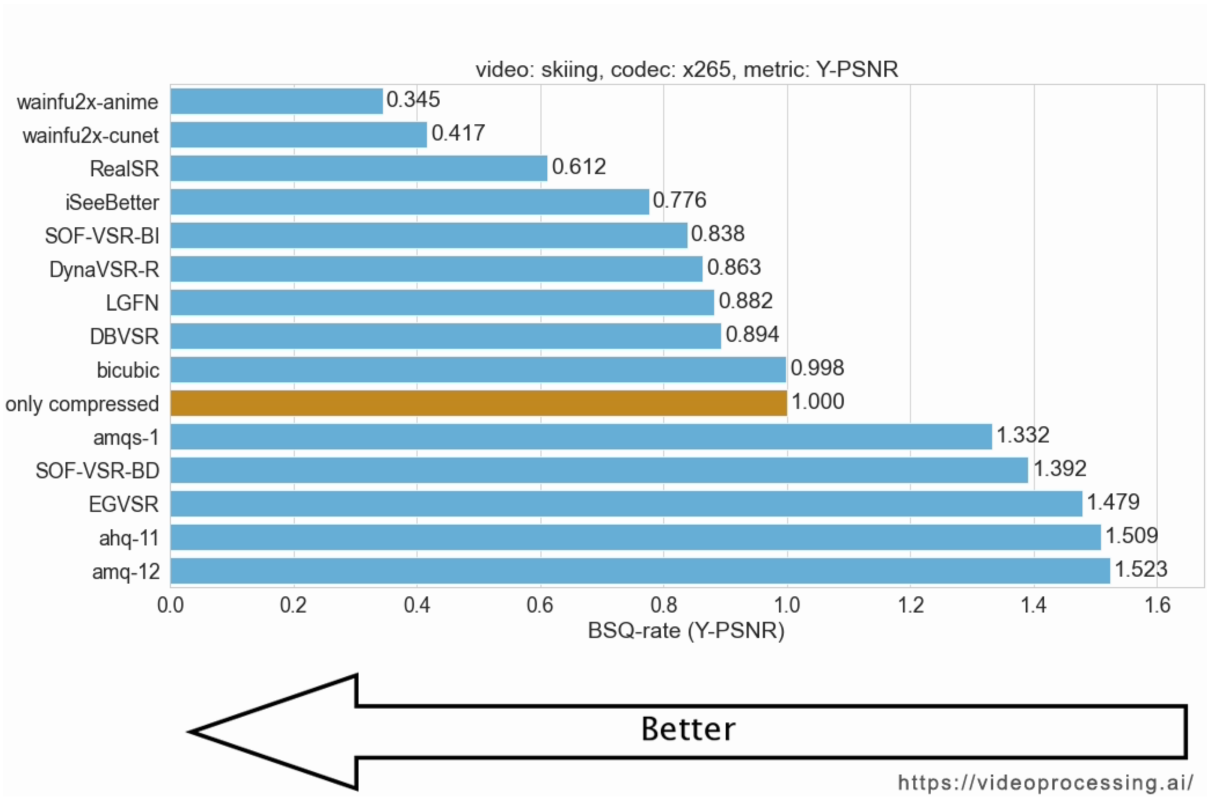


Figure 15b: BSQ-rate relative to “only compressed” —
skiing sequence, x265 codec, Y-PSNR metric

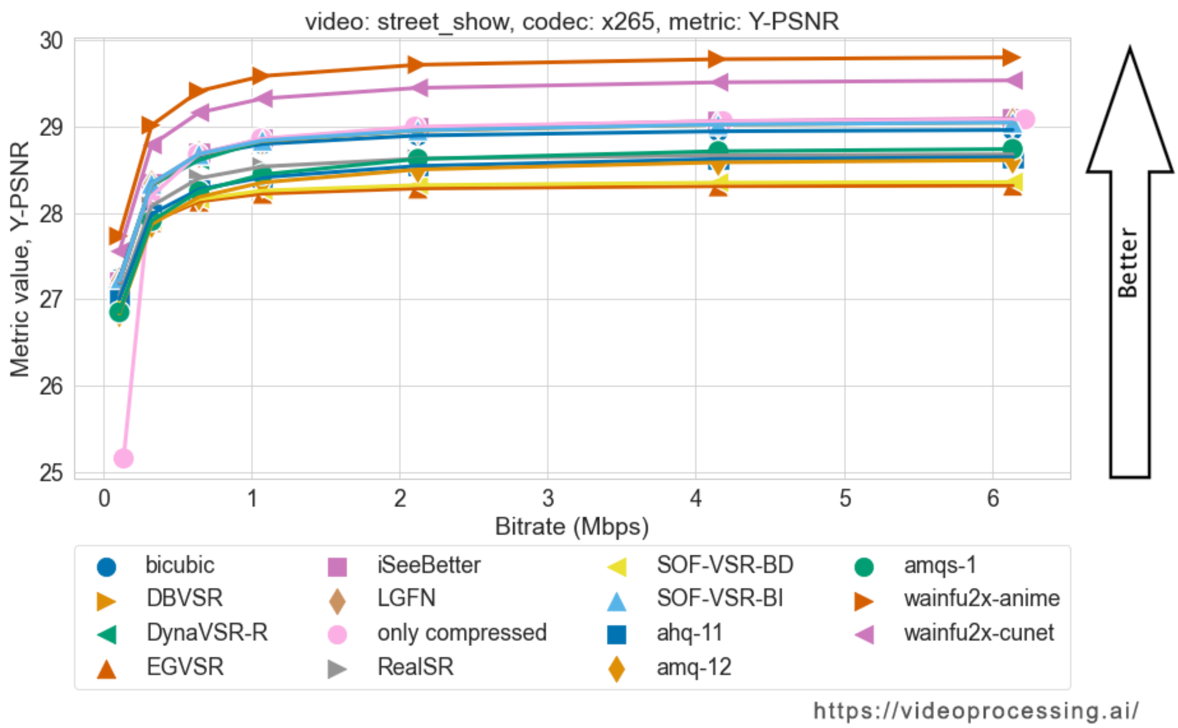


Figure 16a: Bitrate/Quality — *street_show* sequence, x265 codec, Y-PSNR metric

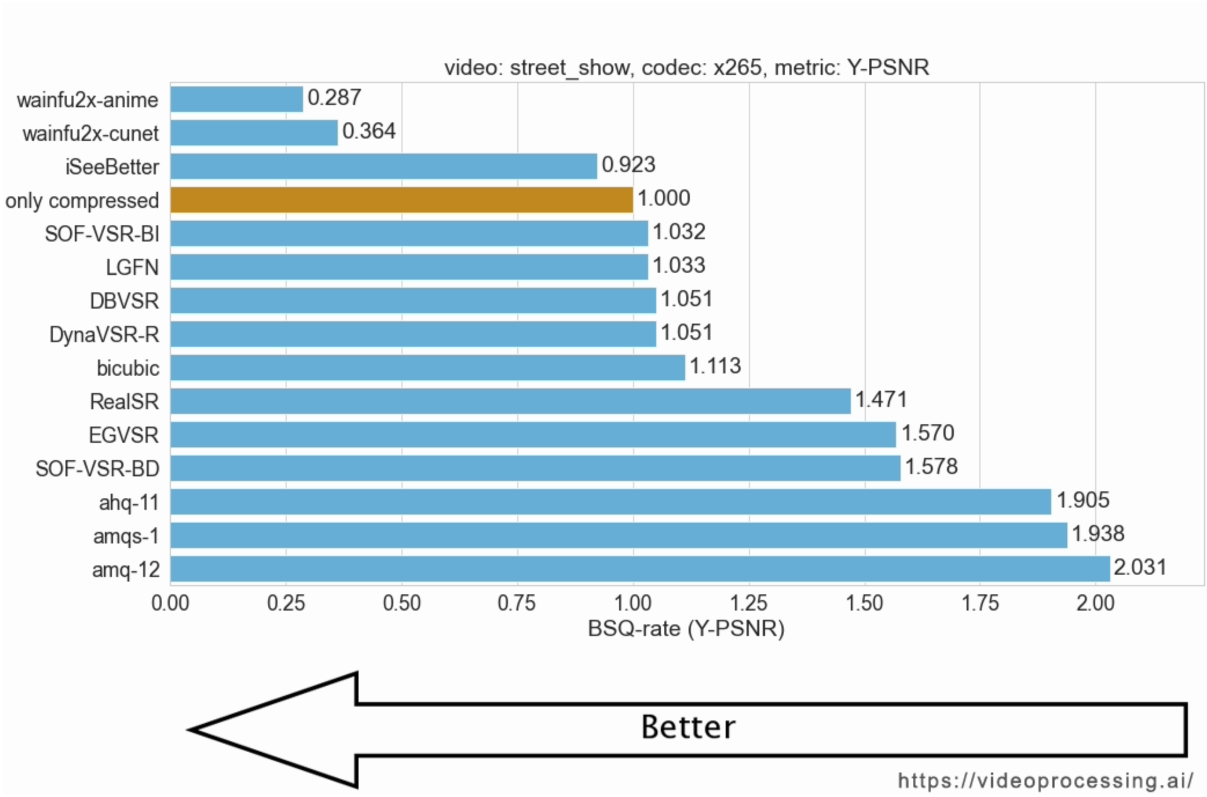


Figure 16b: BSQ-rate relative to “only compressed” — *street_show* sequence, x265 codec, Y-PSNR metric

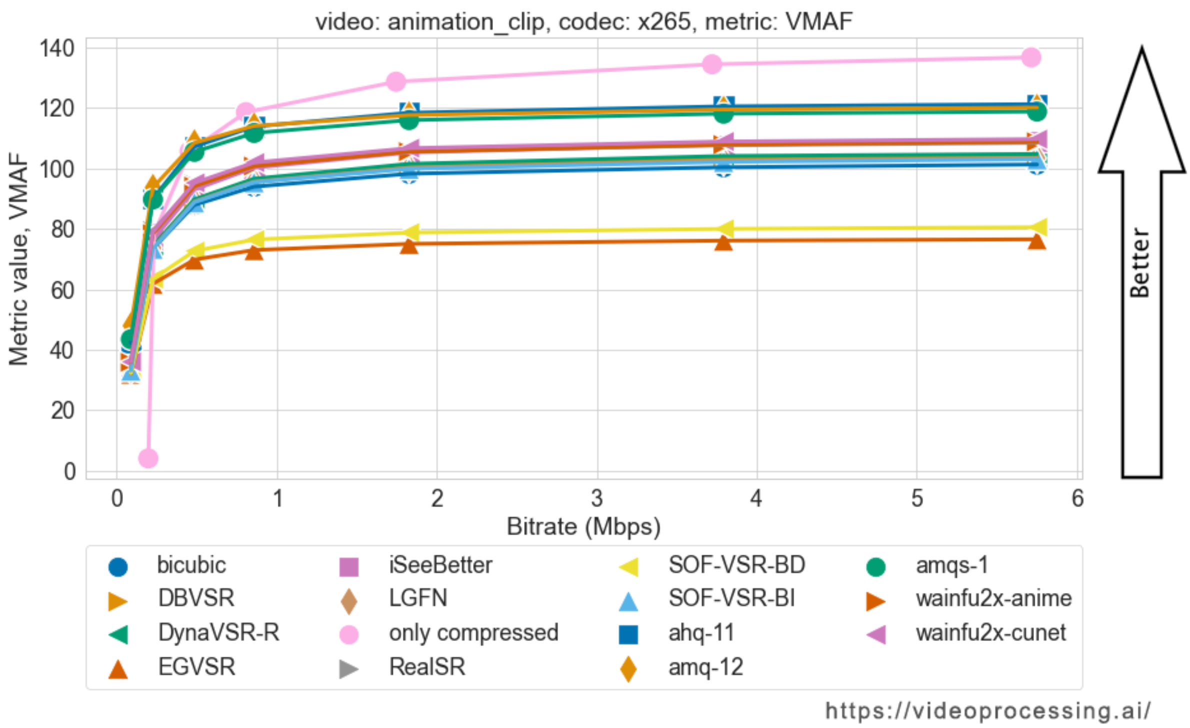


Figure 17a: Bitrate/Quality — *animation_clip* sequence, x265 codec, VMAF metric

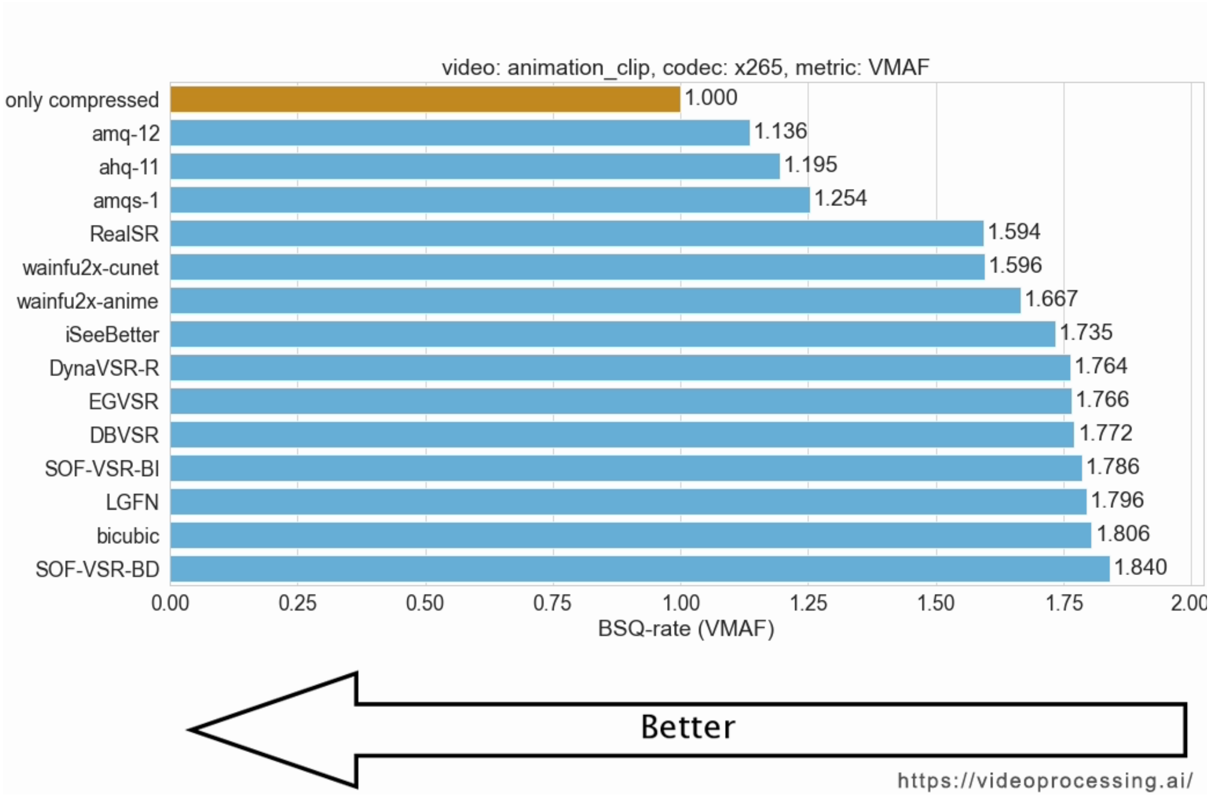


Figure 17b: BSQ-rate relative to “only compressed” — *animation_clip* sequence, x265 codec, VMAF metric

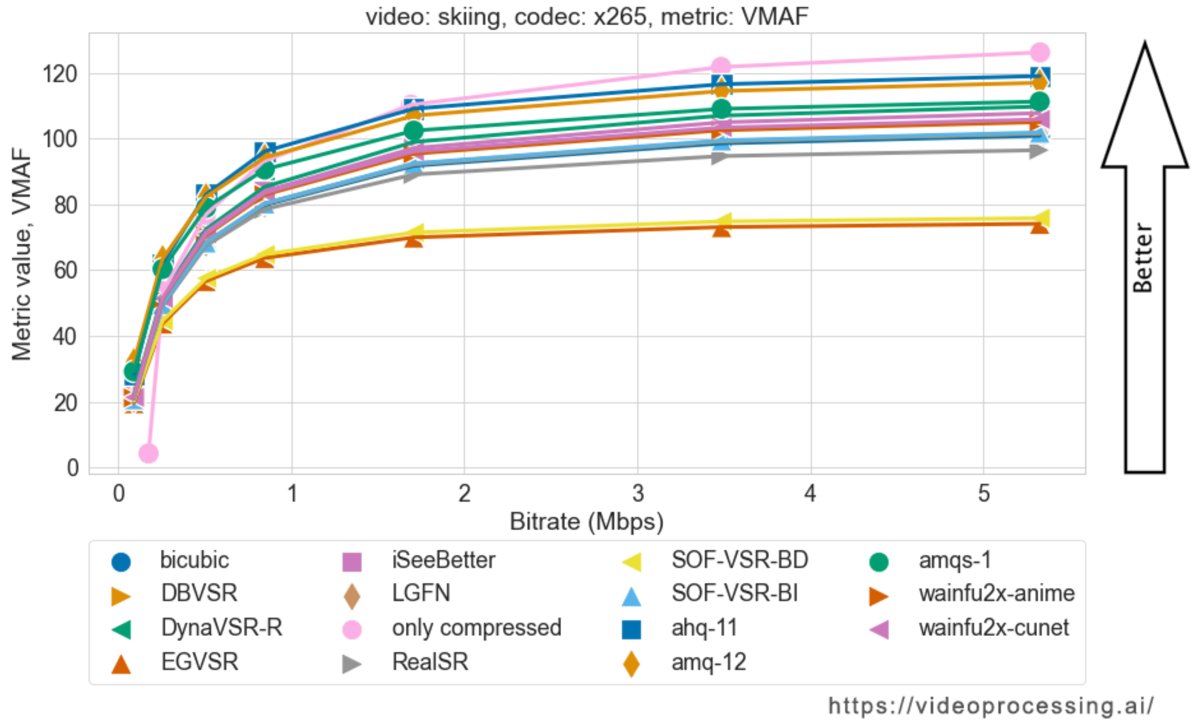


Figure 18a: Bitrate/Quality — *skiing* sequence, x265 codec, VMAF metric

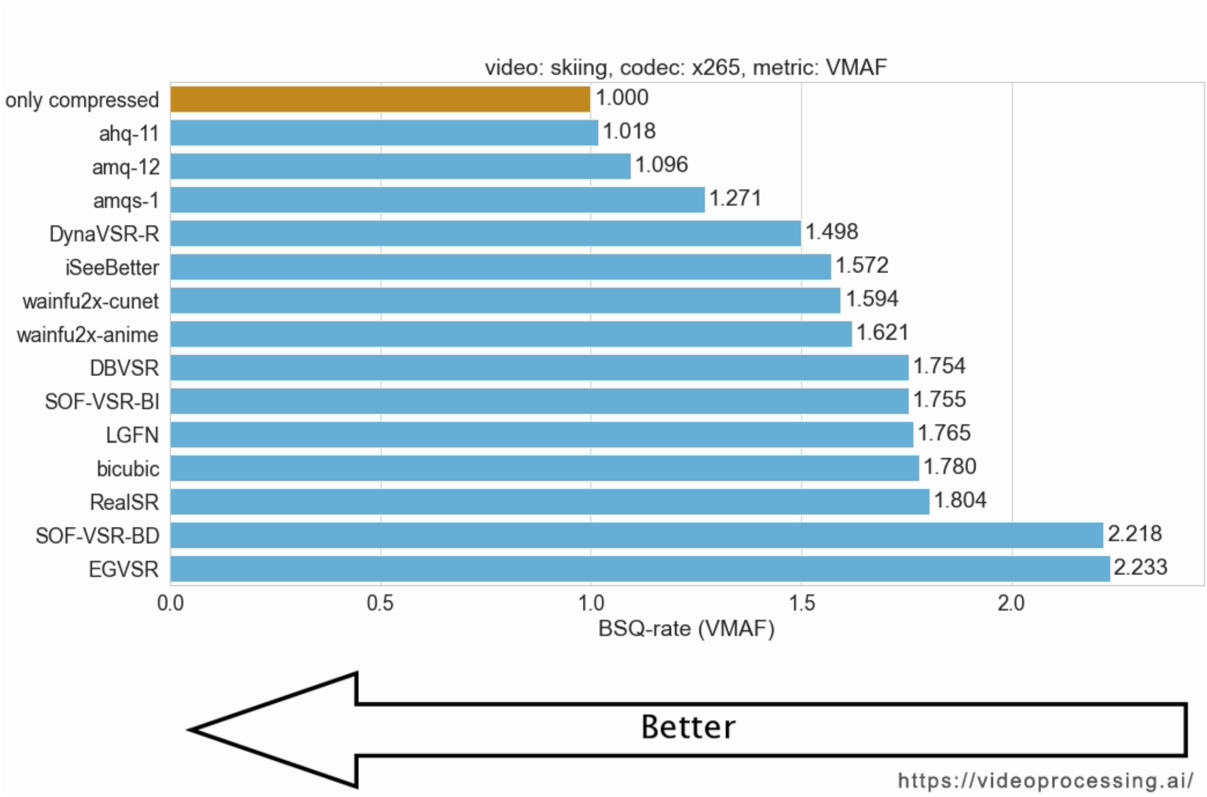


Figure 18b: BSQ-rate relative to “only compressed” —
skiing sequence, x265 codec, VMAF metric

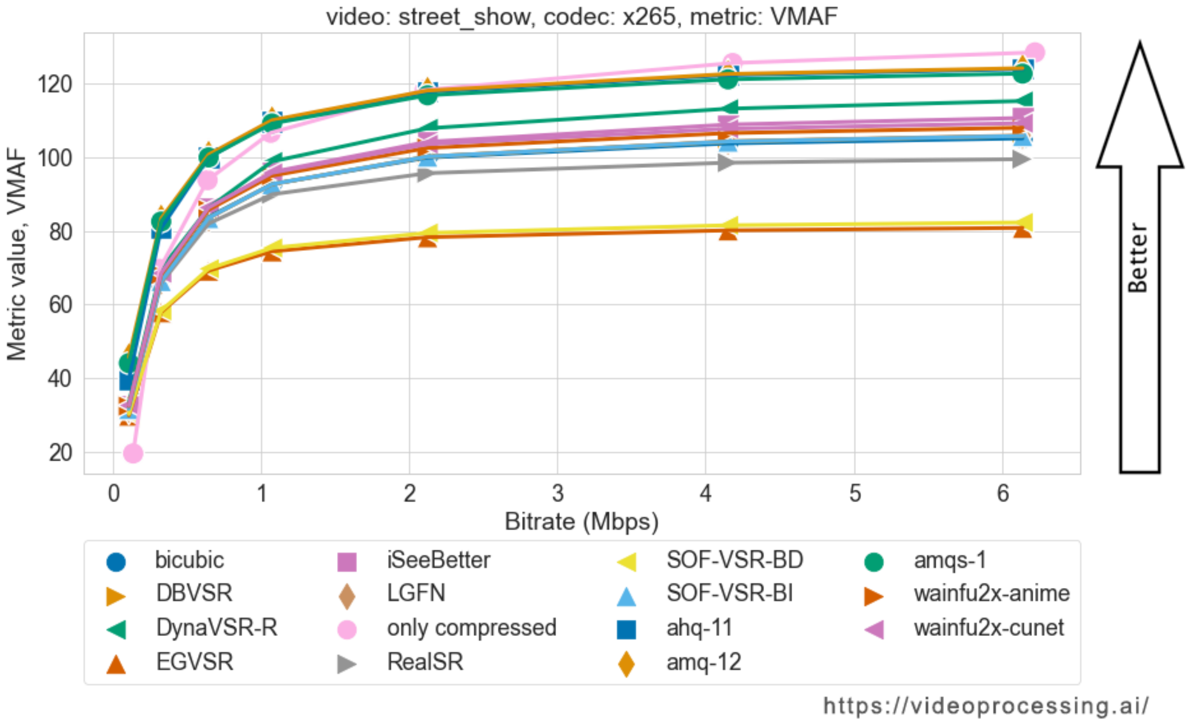


Figure 19a: Bitrate/Quality — *street_show* sequence, x265 codec, VMAF metric

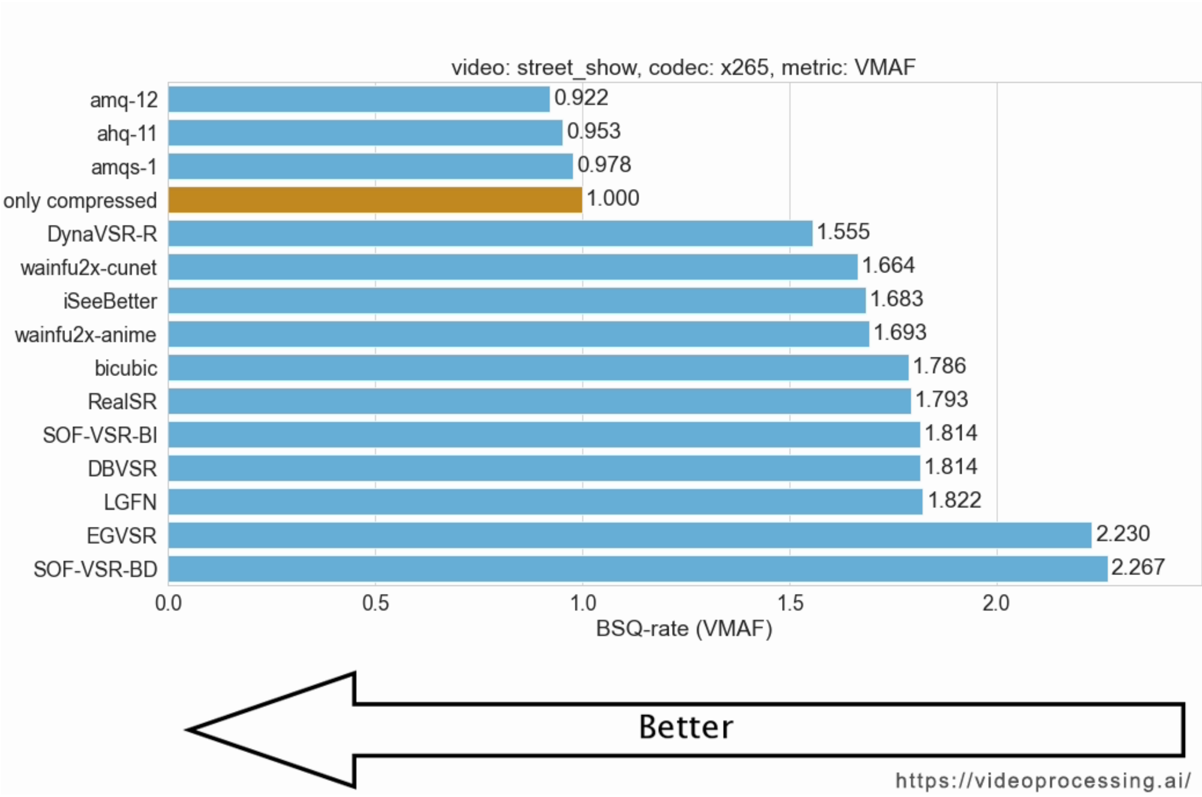


Figure 19b: BSQ-rate relative to "only compressed" —
street_show sequence, x265 codec, VMAF metric

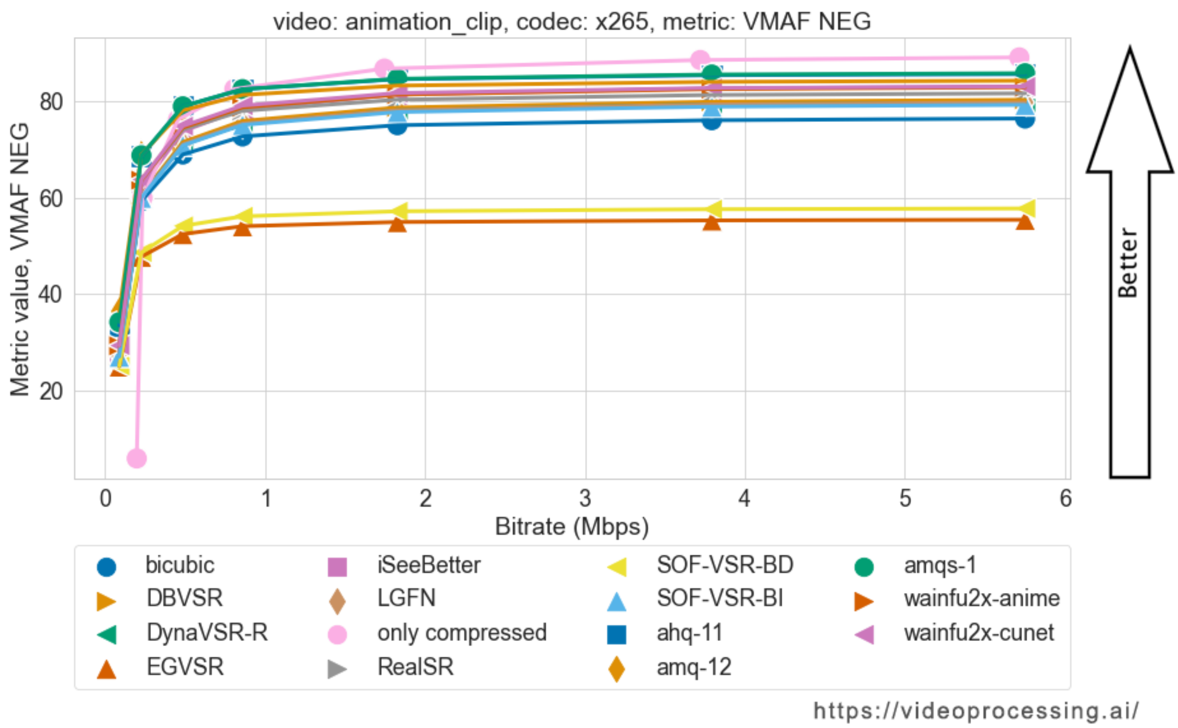


Figure 20a: Bitrate/Quality — *animation_clip* sequence, x265 codec, VMAF NEG metric

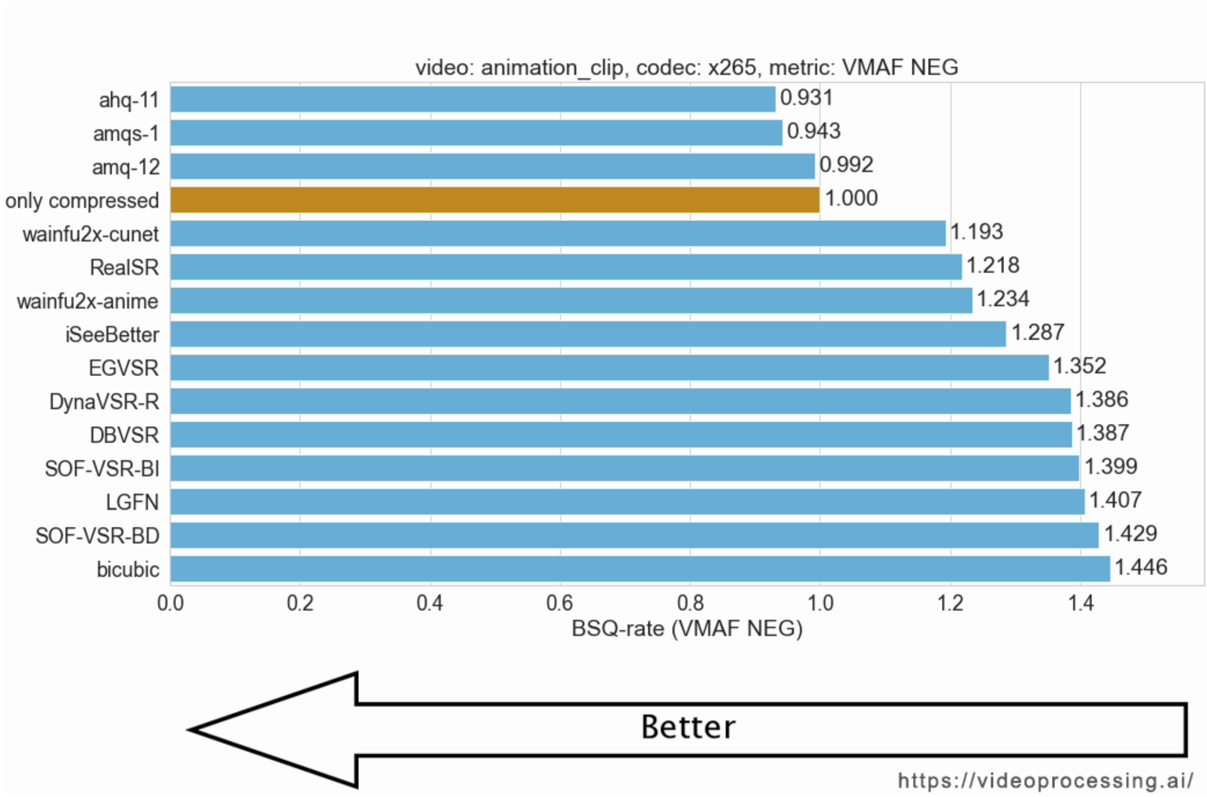


Figure 20b: BSQ-rate relative to "only compressed" — *animation_clip* sequence, x265 codec, VMAF NEG metric

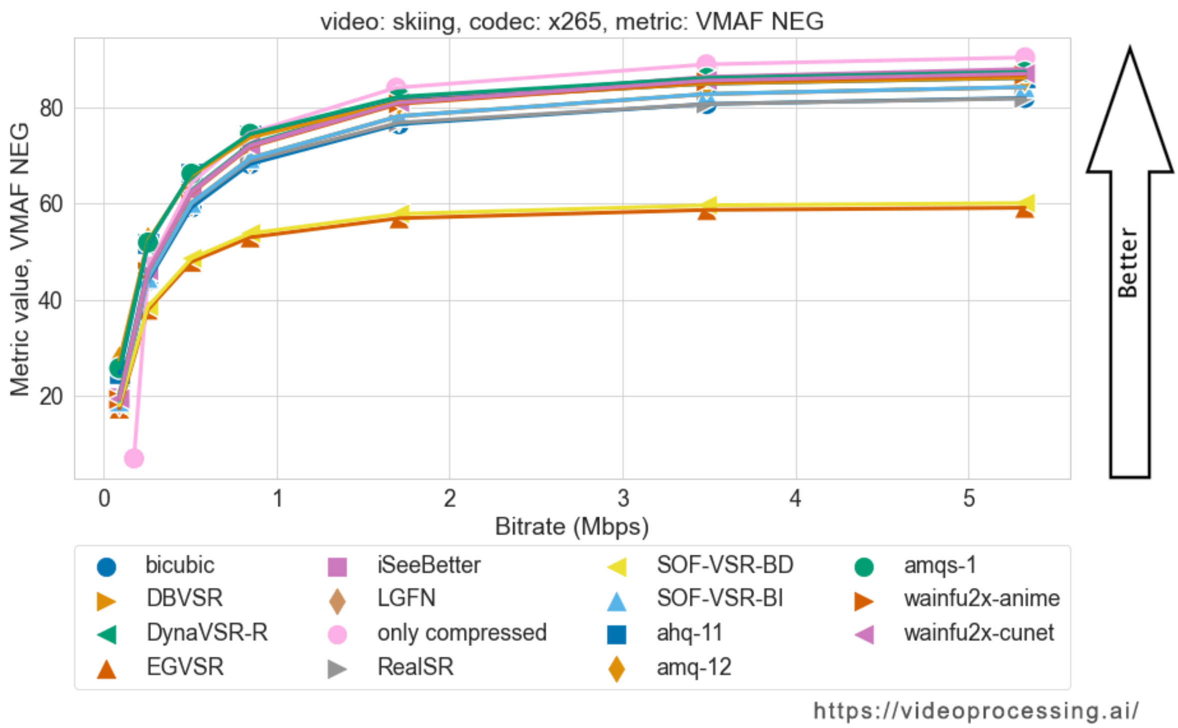


Figure 21a: Bitrate/Quality — *skiing* sequence, x265 codec, VMAF NEG metric

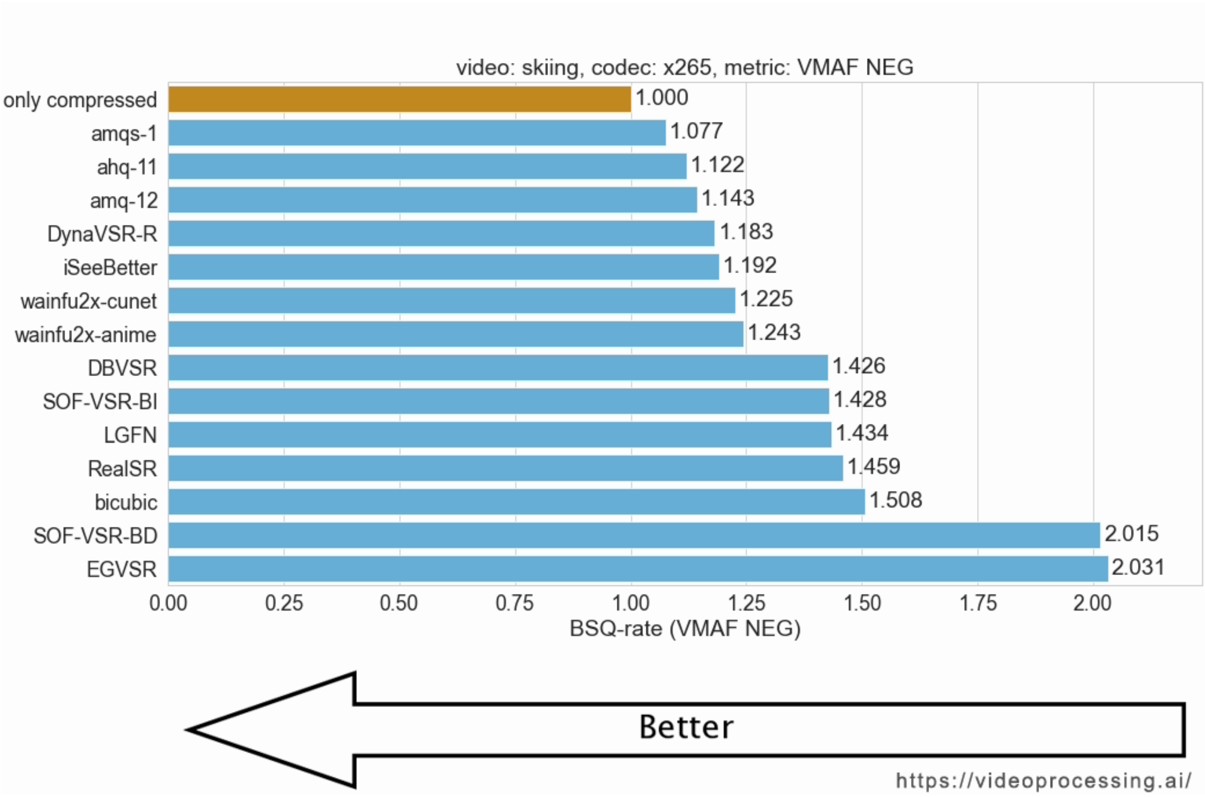


Figure 21b: BSQ-rate relative to “only compressed” —
skiing sequence, x265 codec, VMAF NEG metric

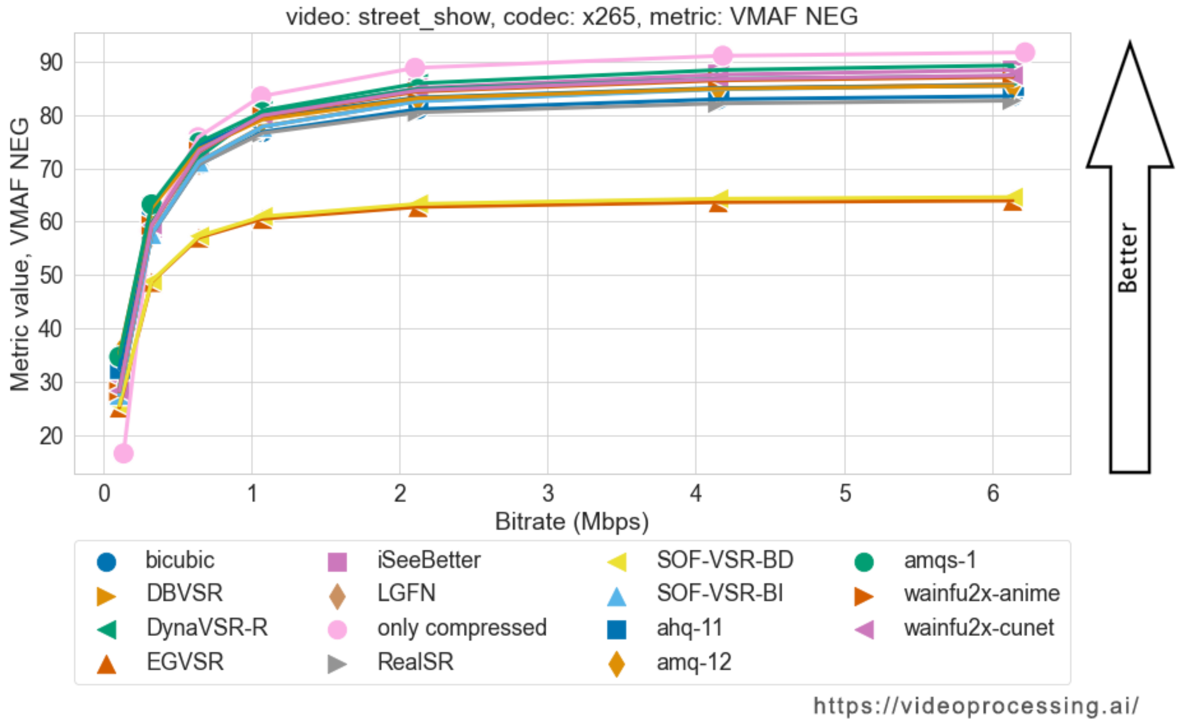


Figure 22a: Bitrate/Quality — *street_show* sequence, x265 codec, VMAF NEG metric

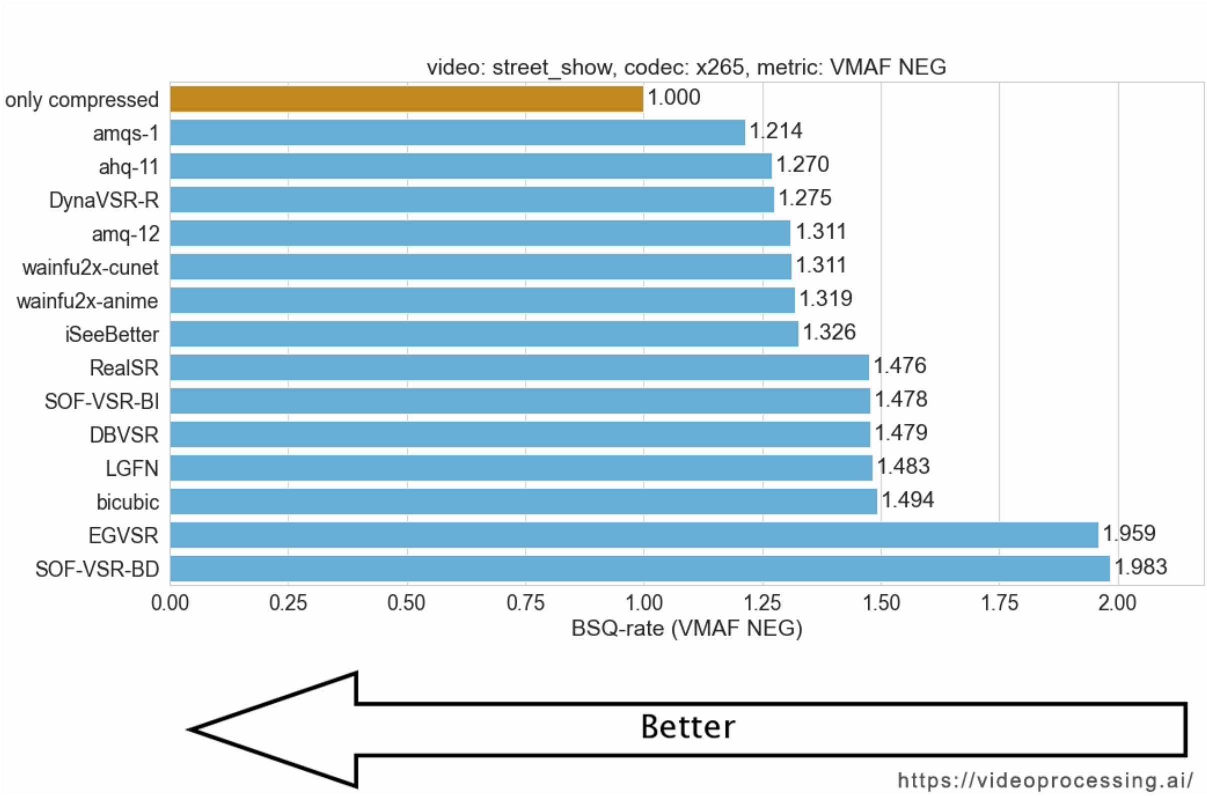


Figure 22b: BSQ-rate relative to “only compressed” —
street_show sequence, x265 codec, VMAF NEG metric

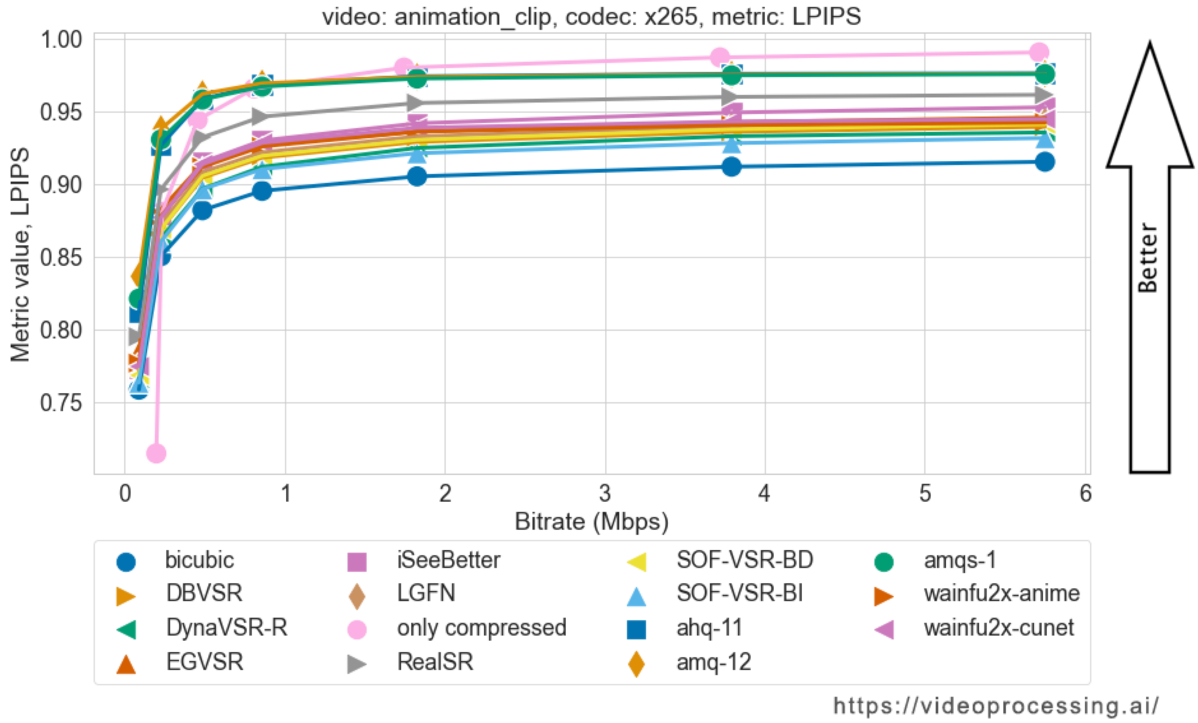


Figure 23a: Bitrate/Quality — *animation_clip* sequence, x265 codec, LPIPS metric

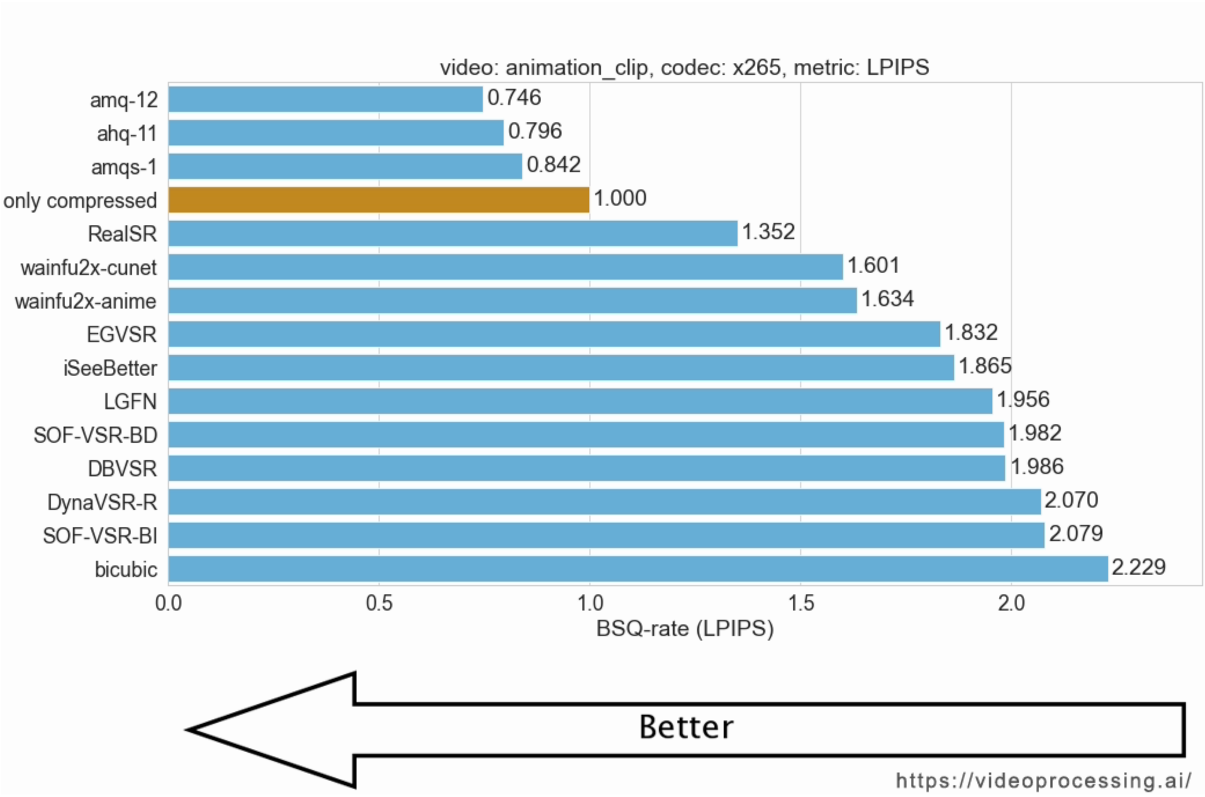


Figure 23b: BSQ-rate relative to “only compressed” — *animation_clip* sequence, x265 codec, LPIPS metric

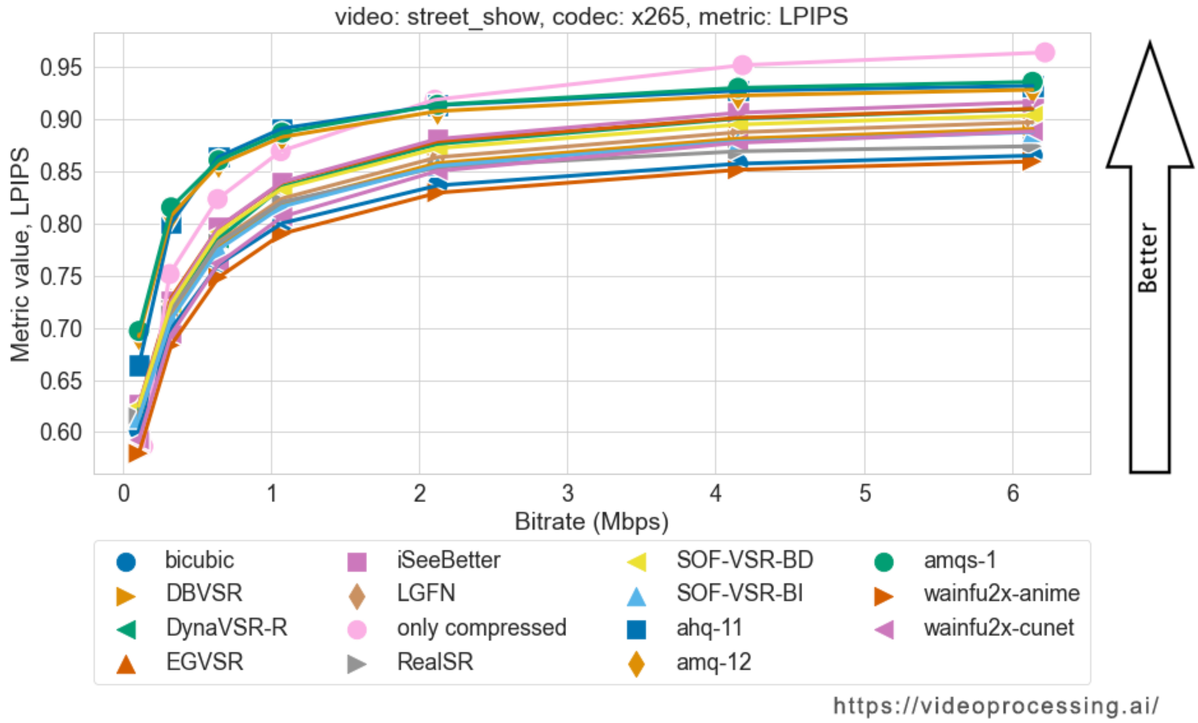


Figure 24a: Bitrate/Quality — *street_show* sequence, x265 codec, LPIPS metric

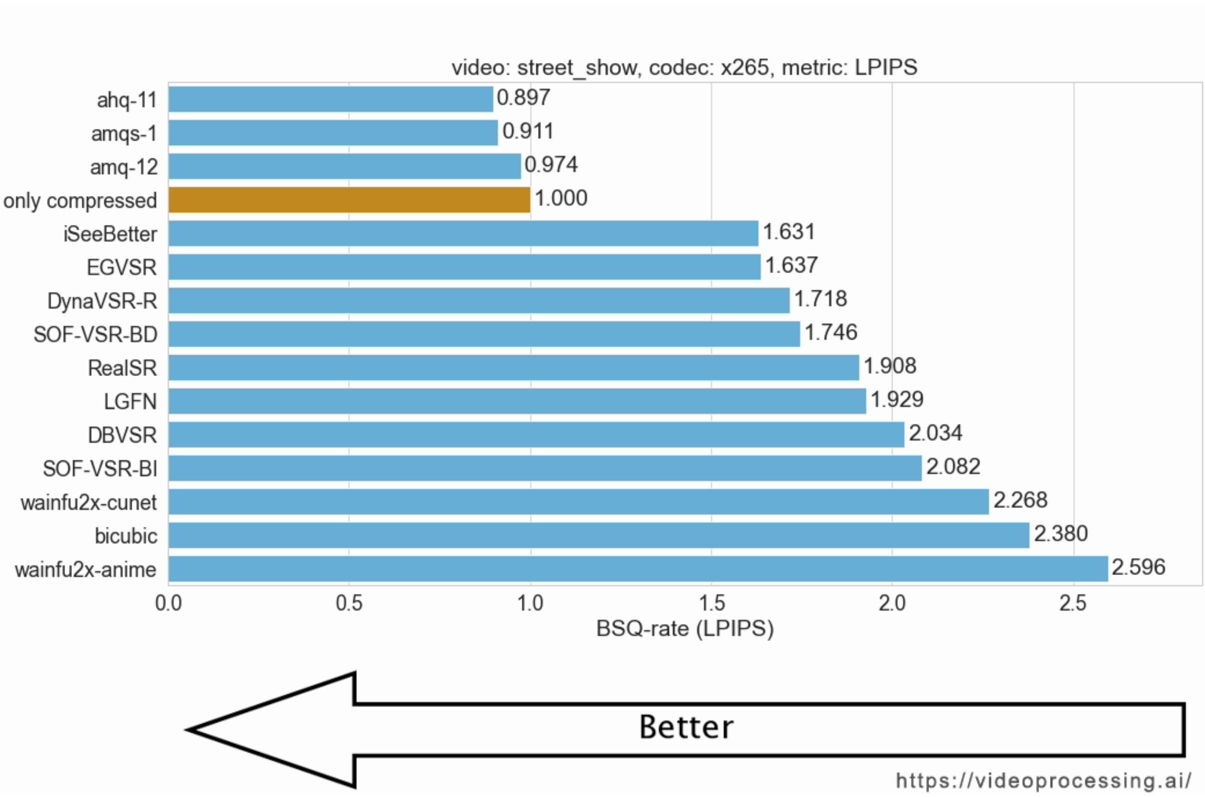


Figure 24b: BSQ-rate relative to “only compressed” —
street_show sequence, x265 codec, LPIPS metric

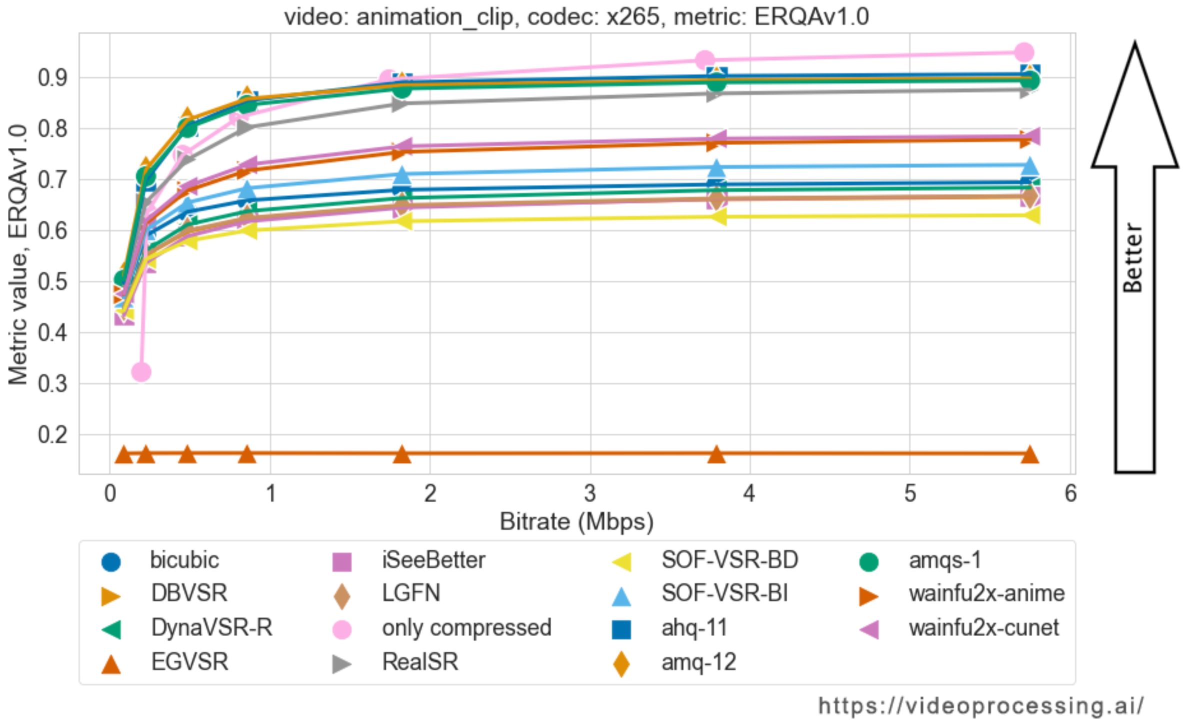


Figure 25a: Bitrate/Quality — *animation_clip* sequence, x265 codec, ERQAv1.0 metric

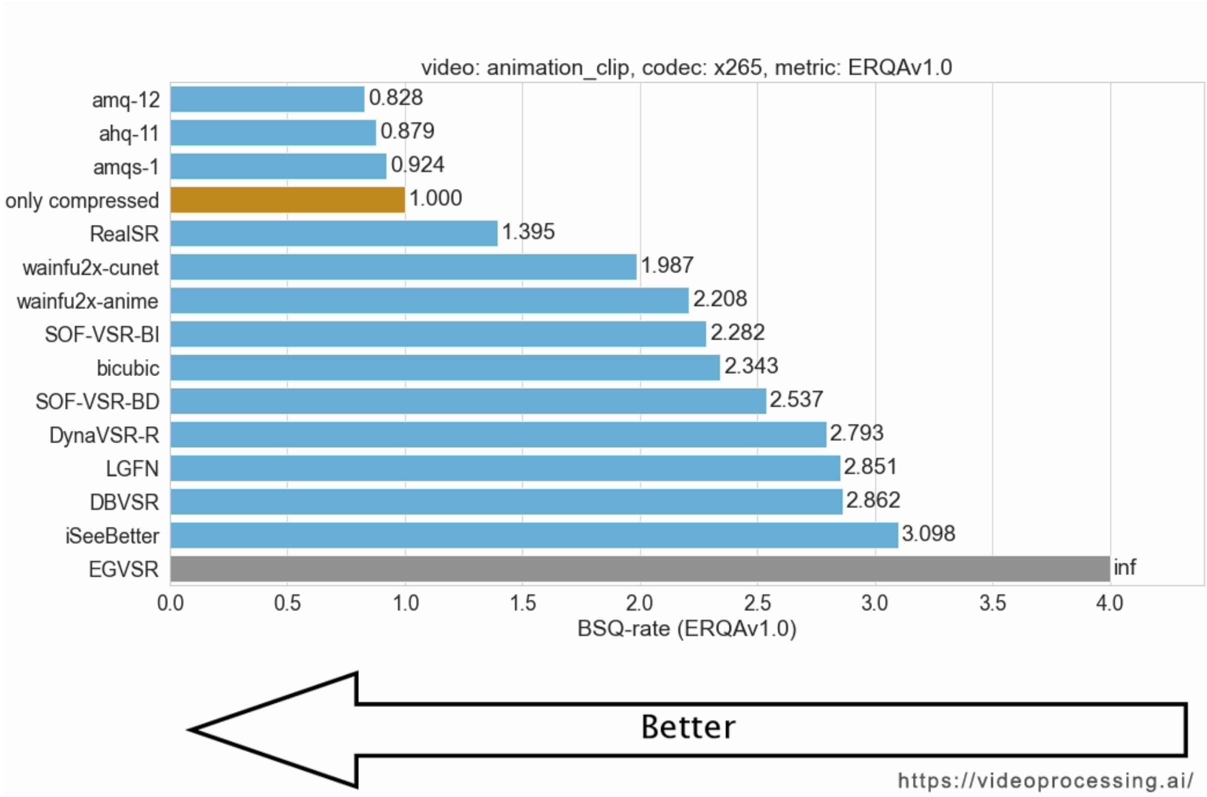


Figure 25b: BSQ-rate relative to “only compressed” —
animation_clip sequence, x265 codec, ERQAv1.0 metric

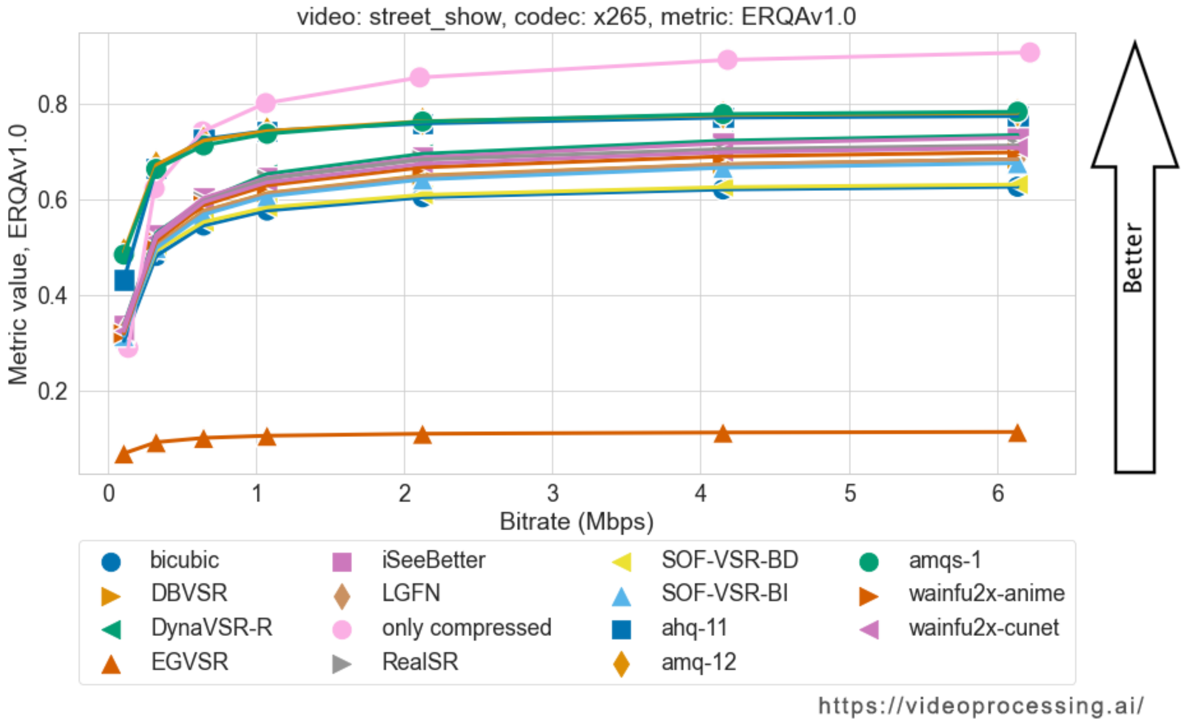


Figure 26a: Bitrate/Quality — *street_show* sequence, x265 codec, ERQAv1.0 metric

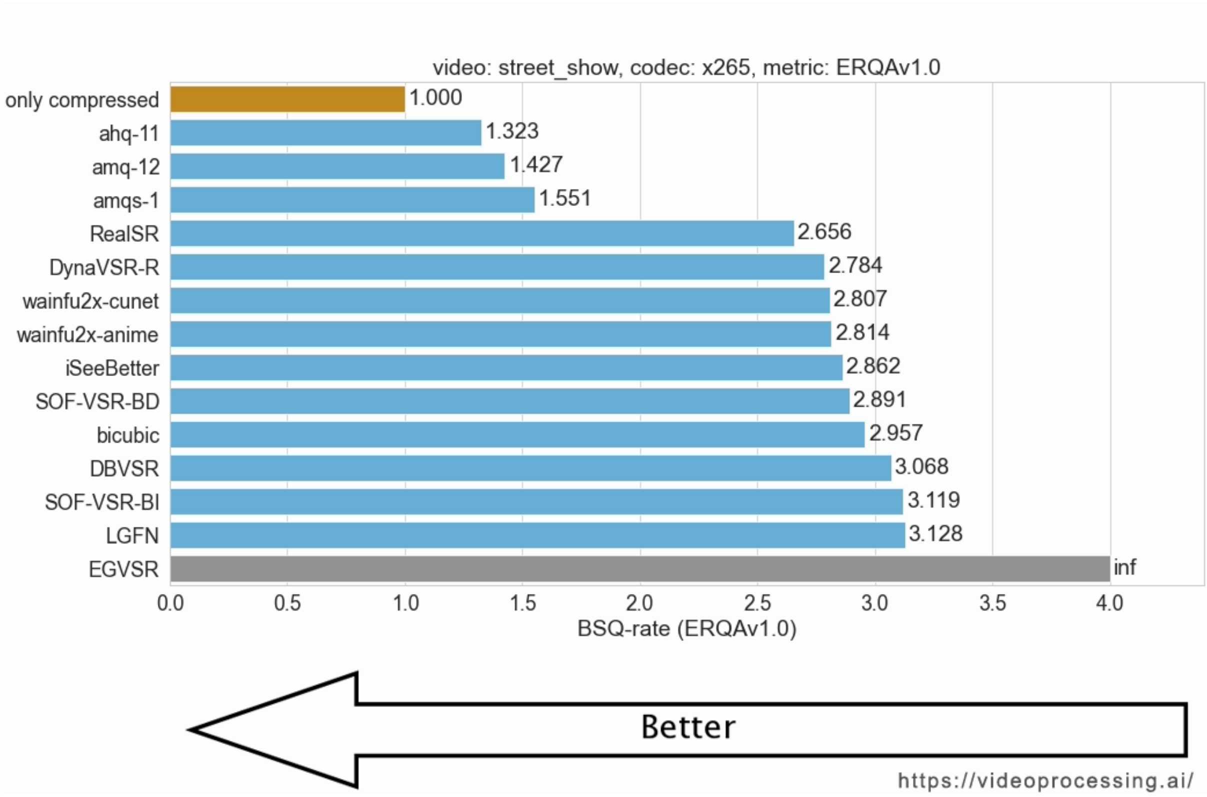


Figure 26b: BSQ-rate relative to “only compressed” —
street_show sequence, x265 codec, ERQAv1.0 metric

In Figure 27 you can see the average BSQ-rate over each metric for the x265 codec. “Only compressed” made by x264 codec was used as a reference.

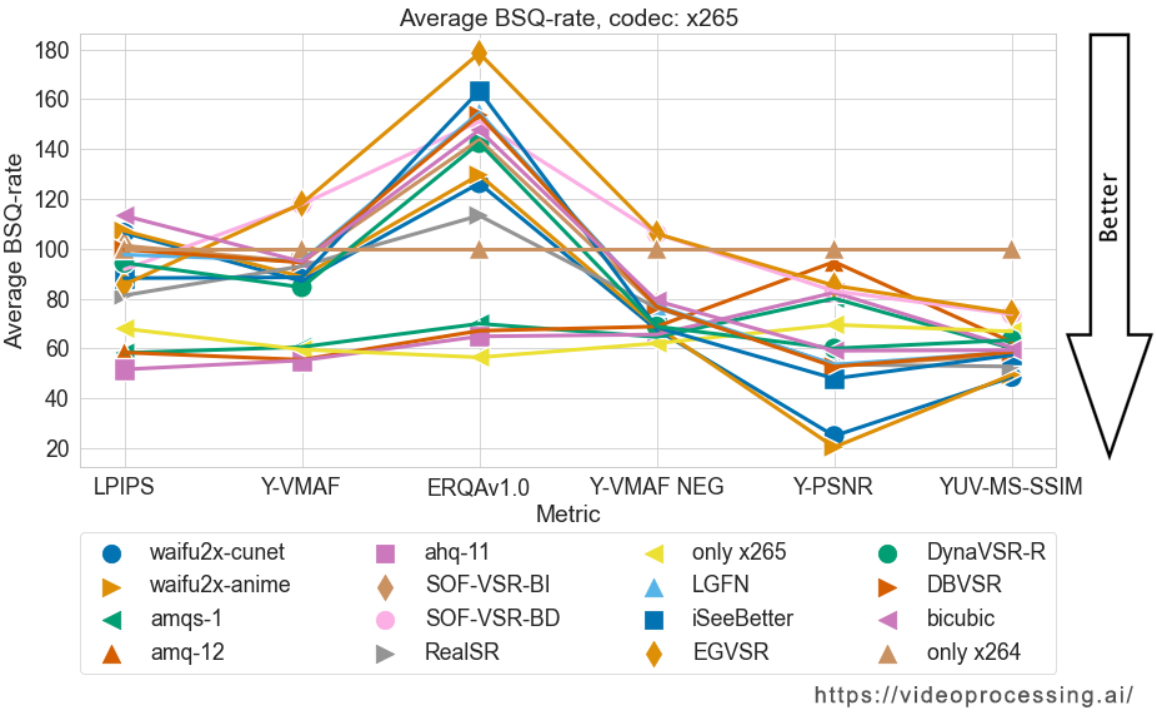


Figure 27: Average BSQ-rate relative to “only x264”.
SR input was compressed with the x265 codec

2.4. aomenc results

In this section, you can see the results of applying SR models on videos compressed with the aomenc codec.

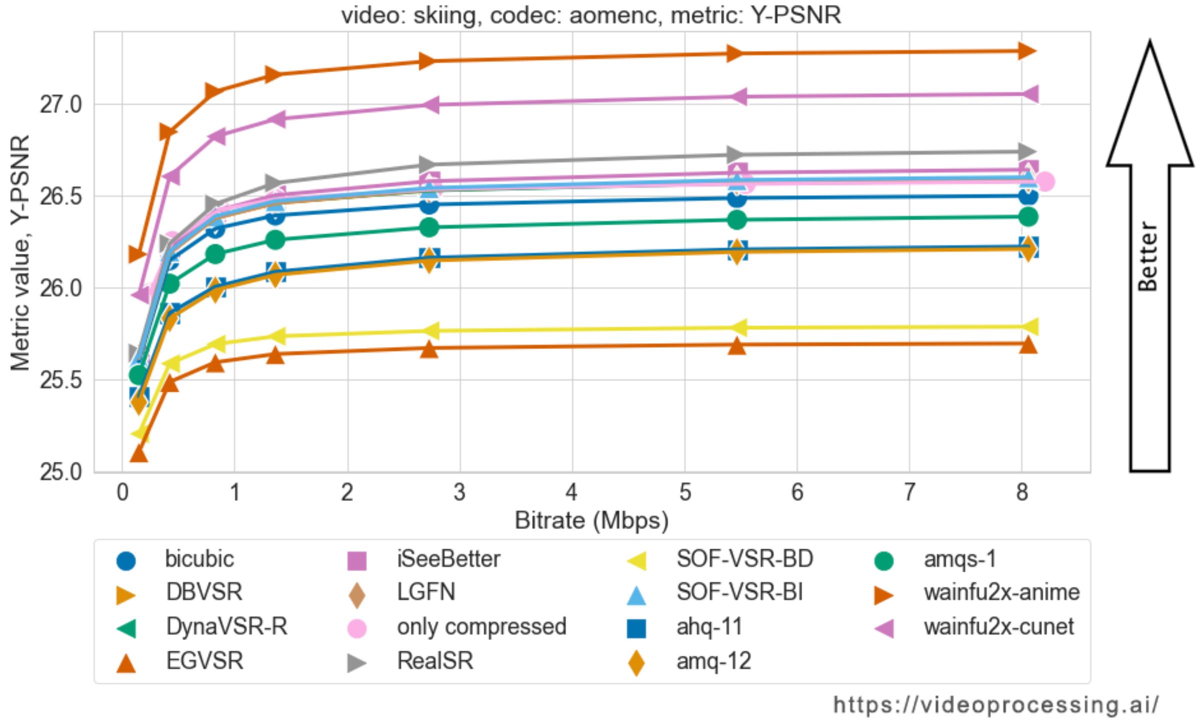


Figure 28a: Bitrate/Quality — *skiing* sequence, aomenc codec, Y-PSNR metric

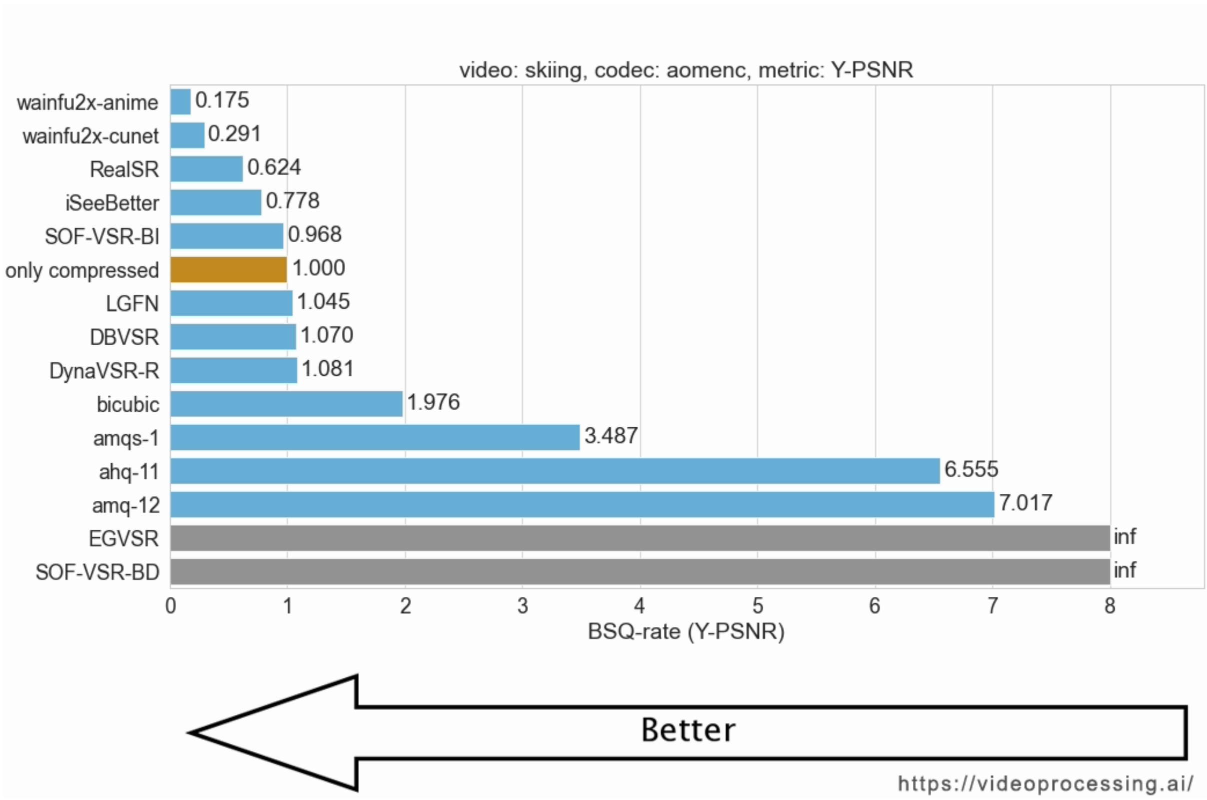


Figure 28b: BSQ-rate relative to "only compressed" —
skiing sequence, aomenc codec, Y-PSNR metric

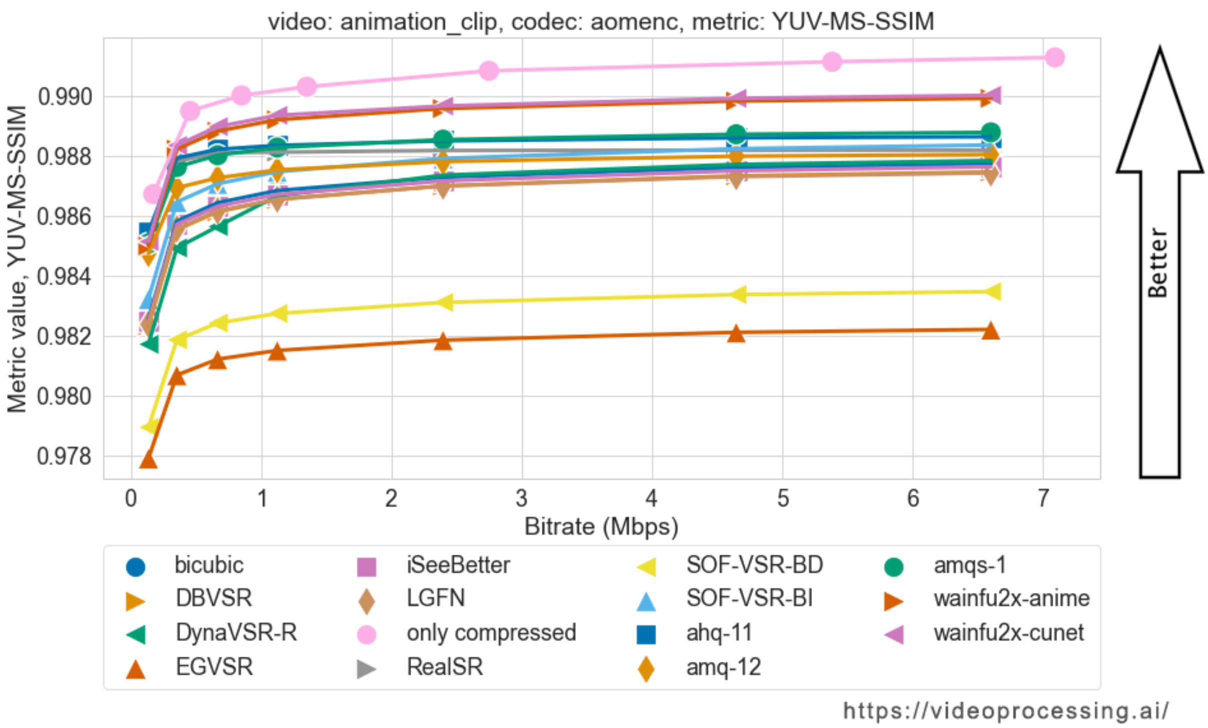


Figure 29a: Bitrate/Quality — *animation_clip* sequence, aomenc codec, YUV-MS-SSIM metric

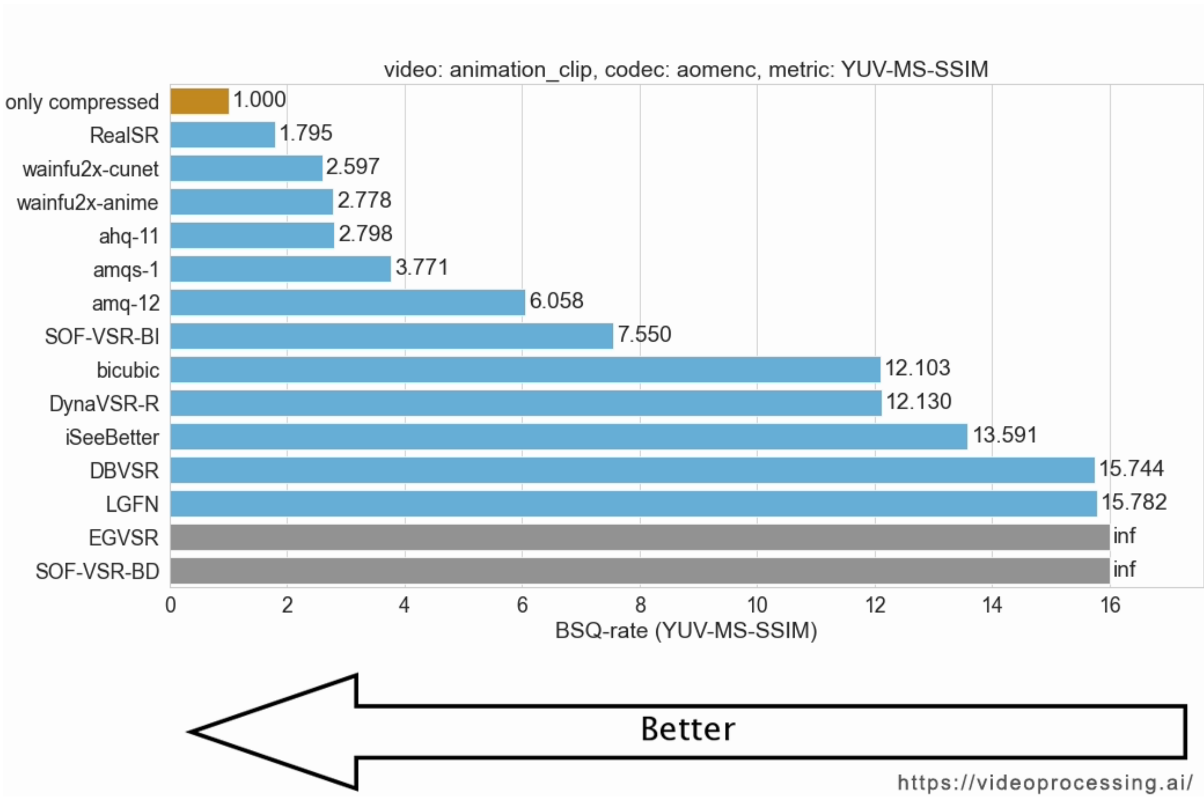


Figure 29b: BSQ-rate relative to “only compressed” —
animation_clip sequence, aomenc codec, YUV-MS-SSIM metric

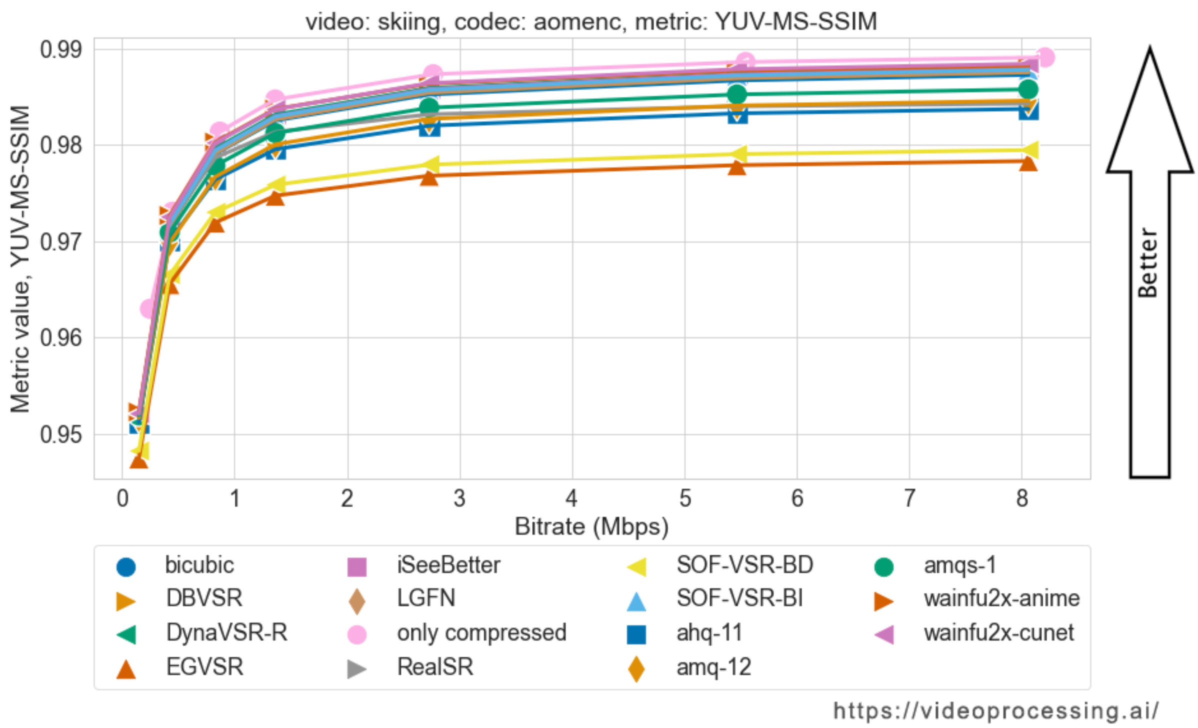


Figure 30a: Bitrate/Quality — *skiing* sequence, aomenc codec, YUV-MS-SSIM metric

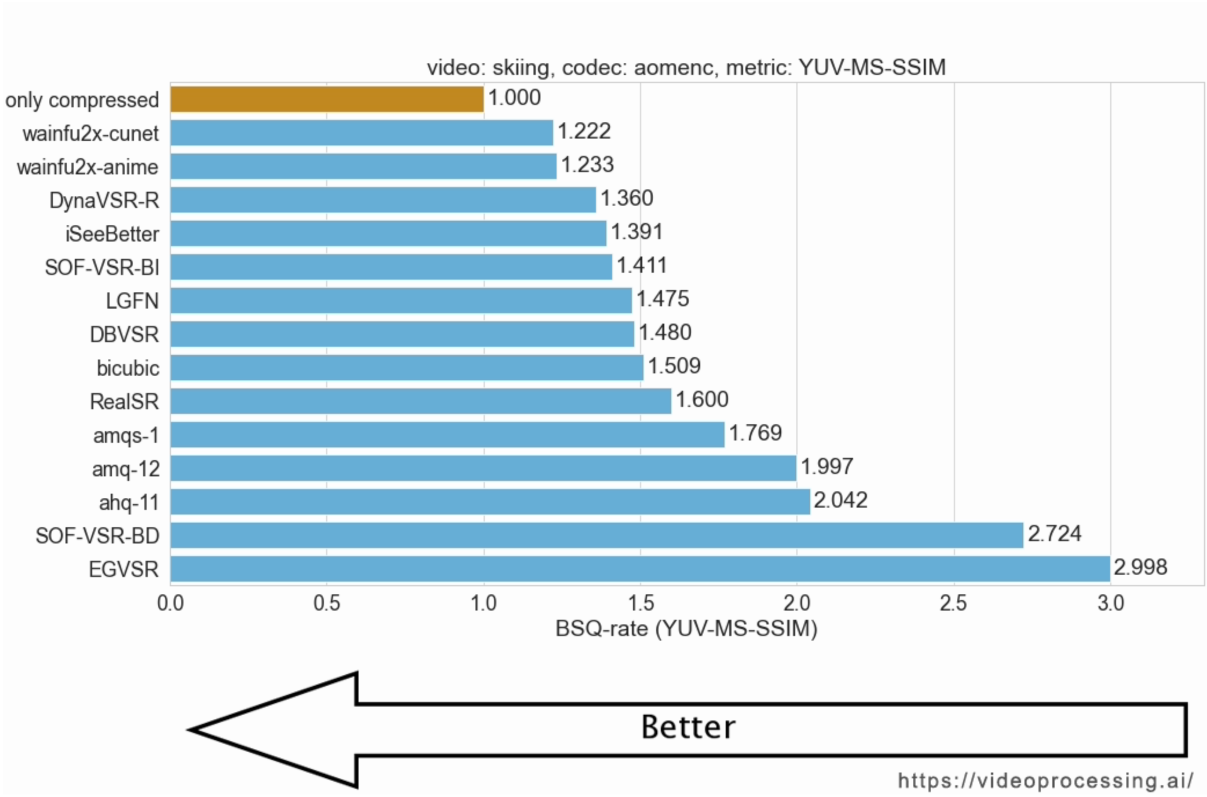


Figure 30b: BSQ-rate relative to "only compressed" —
skiing sequence, aomenc codec, YUV-MS-SSIM metric

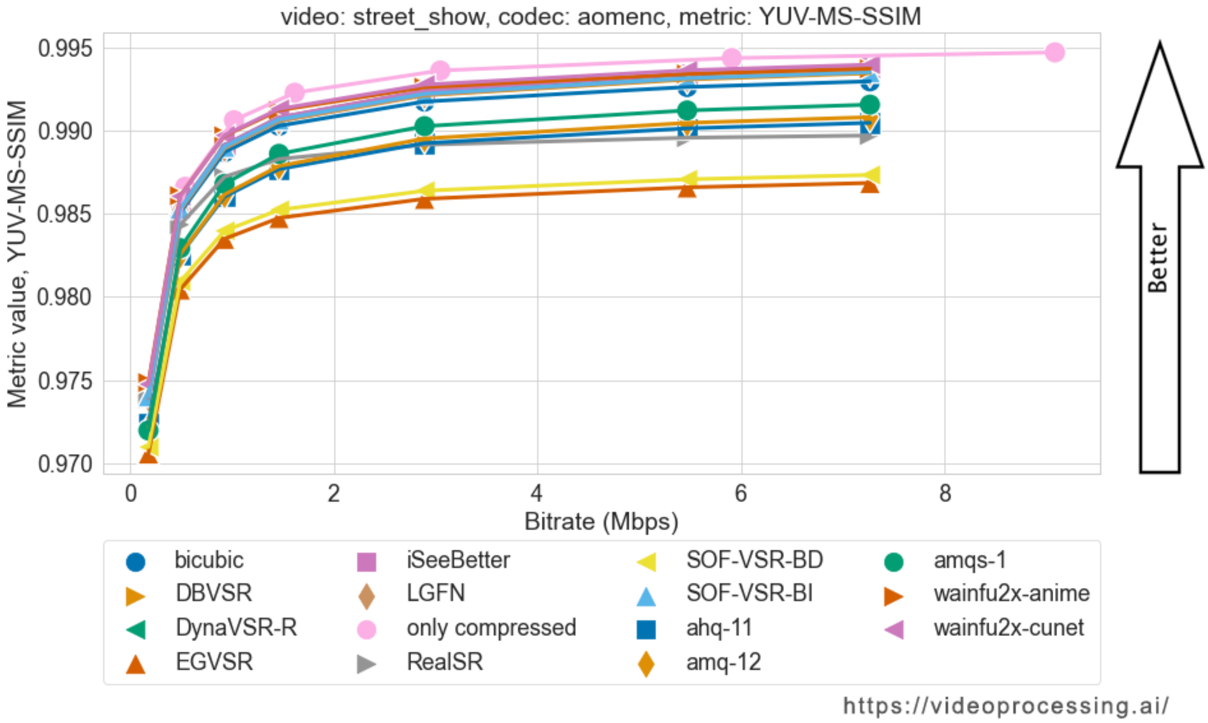


Figure 31a: Bitrate/Quality — *street_show* sequence, aomenc codec, YUV-MS-SSIM metric

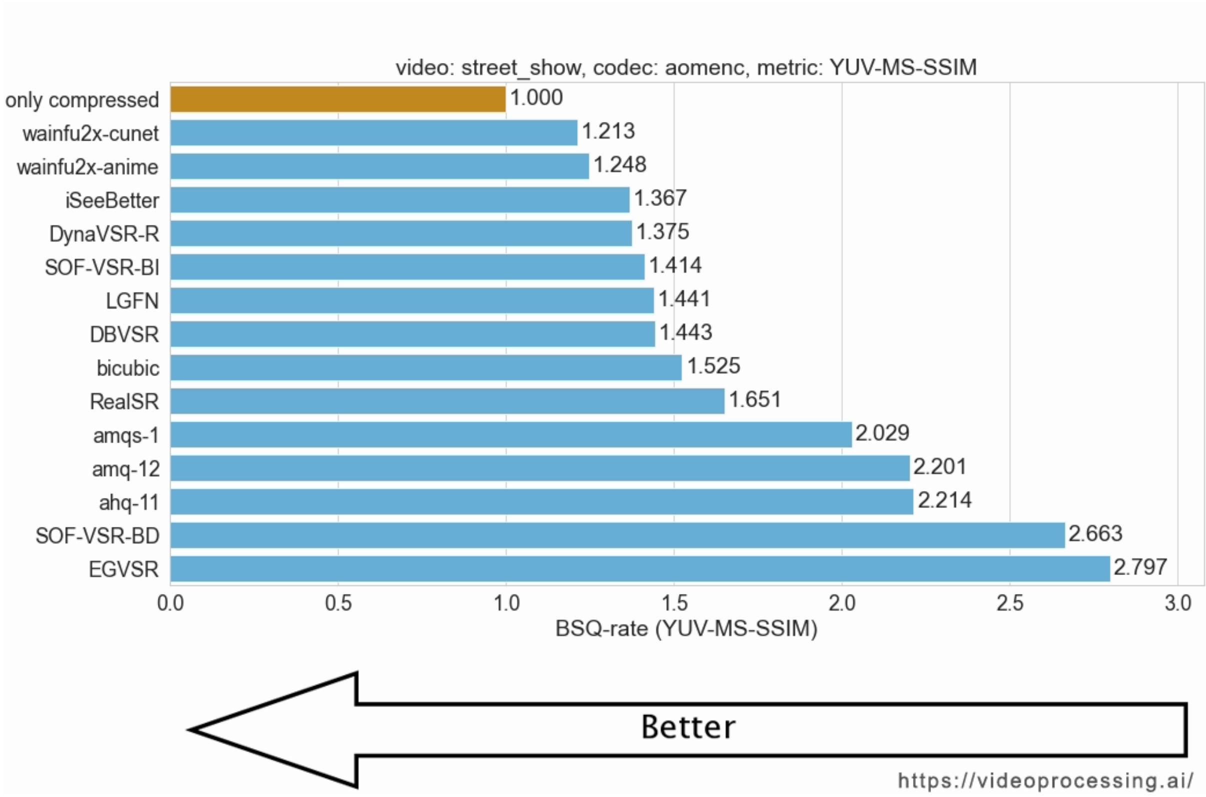


Figure 31b: BSQ-rate relative to “only compressed” —
street_show sequence, aomenc codec, YUV-MS-SSIM metric

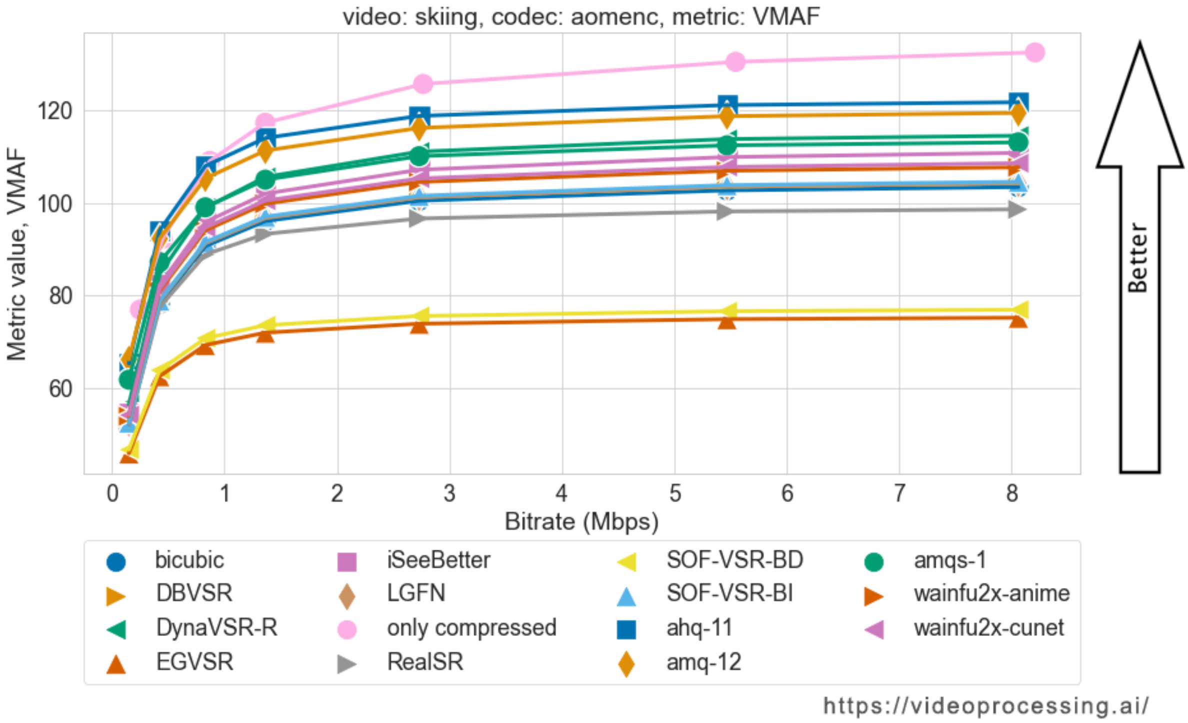


Figure 32a: Bitrate/Quality — *skiing* sequence, aomenc codec, VMAF metric

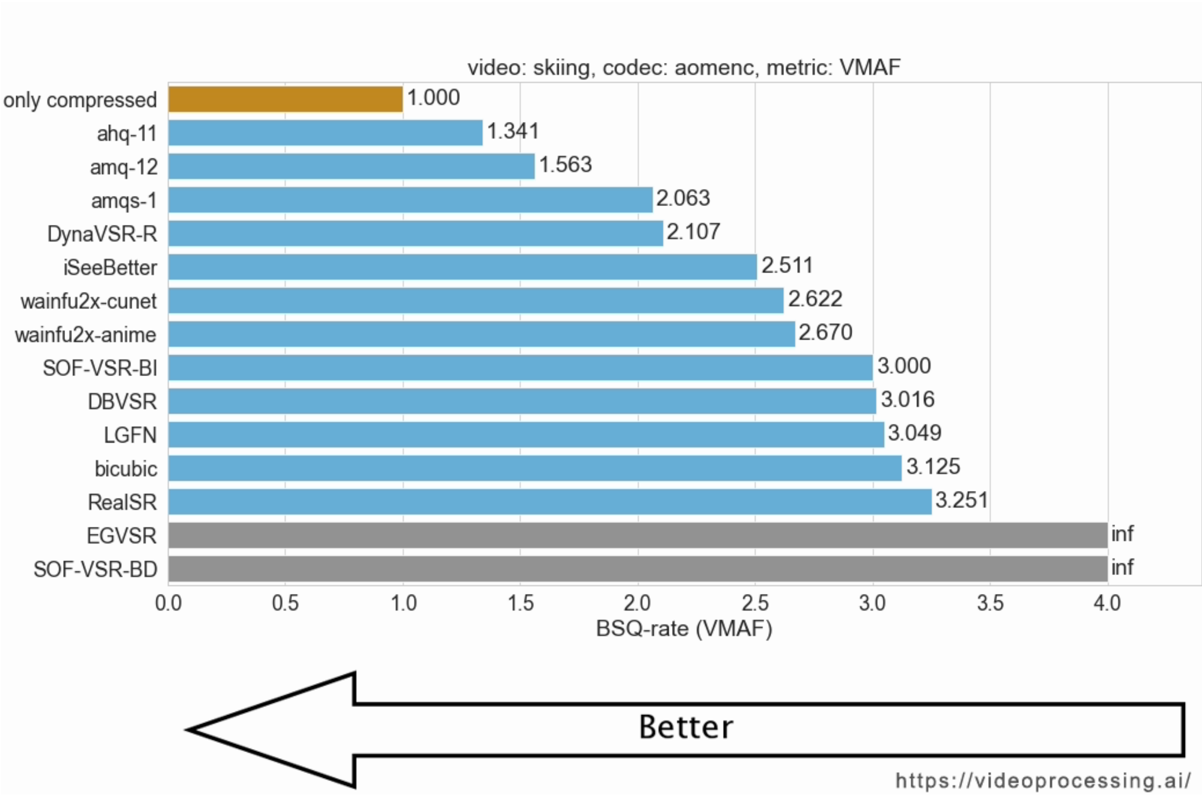


Figure 32b: BSQ-rate relative to “only compressed” —
skiing sequence, aomenc codec, VMAF metric

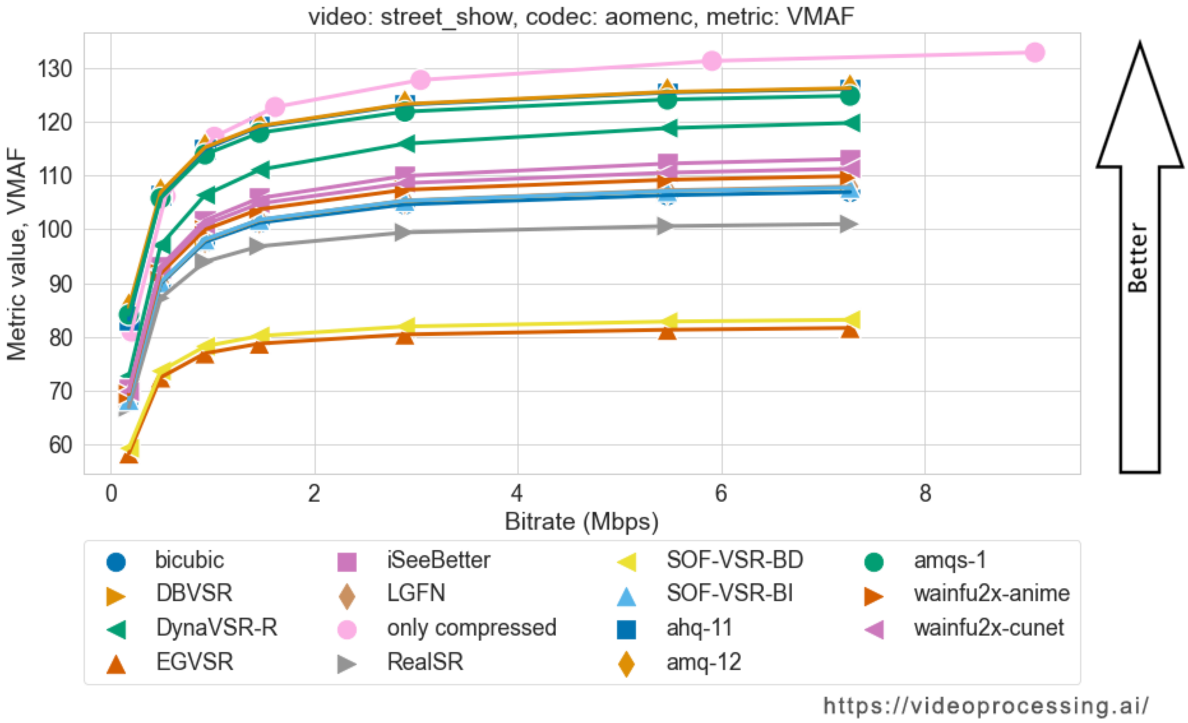


Figure 33a: Bitrate/Quality — *street_show* sequence, aomenc codec, VMAF metric

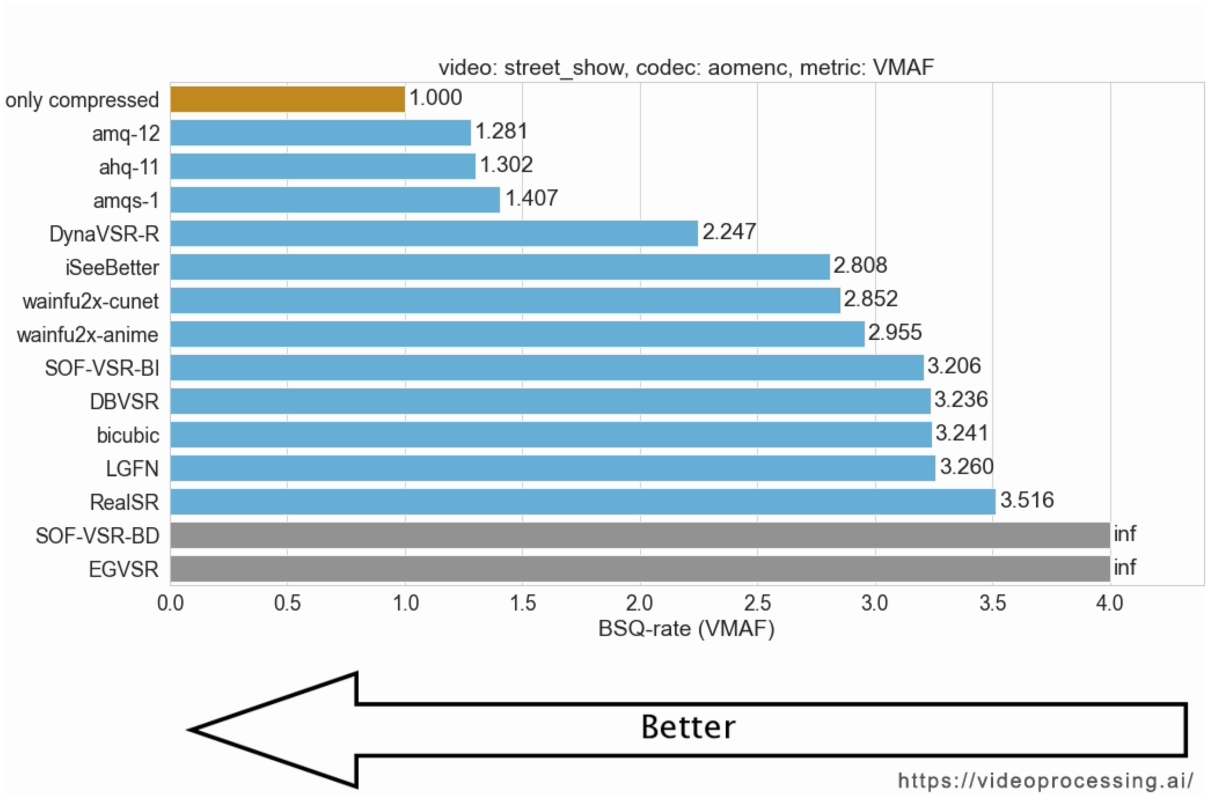


Figure 33b: BSQ-rate relative to “only compressed” —
street_show sequence, aomenc codec, VMAF metric

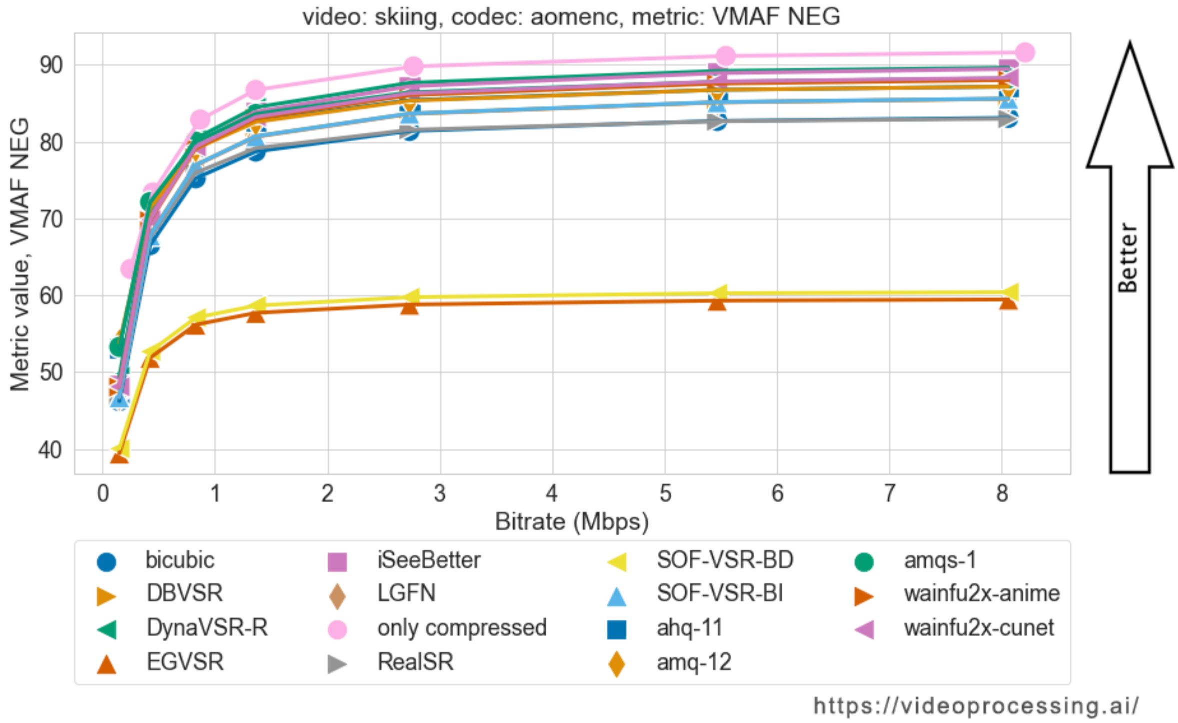


Figure 34a: Bitrate/Quality — *skiing* sequence, aomenc codec, VMAF NEG metric

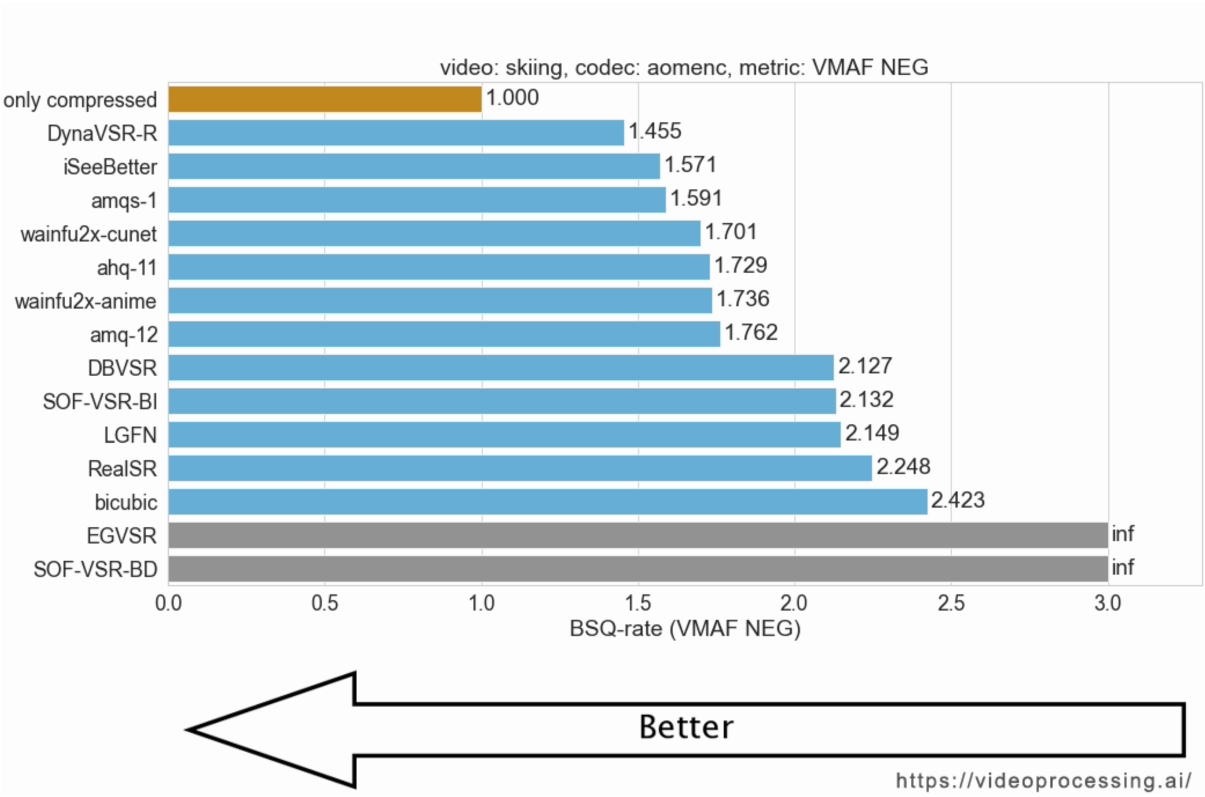


Figure 34b: BSQ-rate relative to “only compressed” —
skiing sequence, aomenc codec, VMAF NEG metric

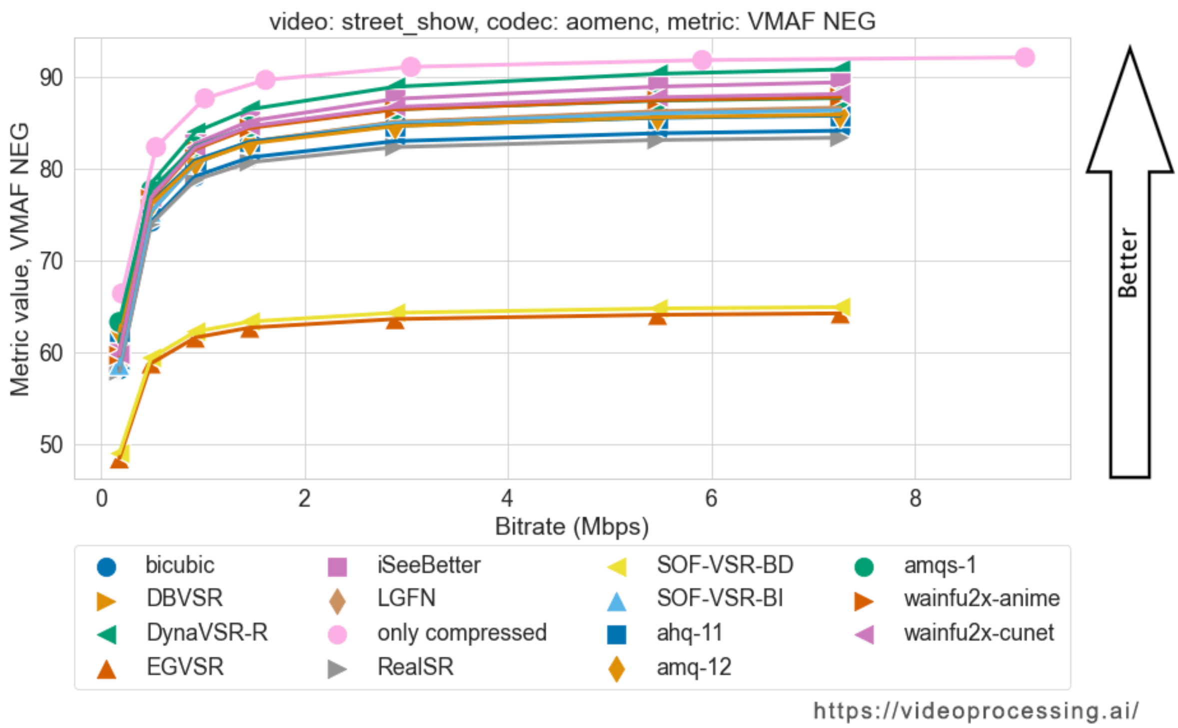


Figure 35a: Bitrate/Quality — *street_show* sequence, aomenc codec, VMAF NEG metric

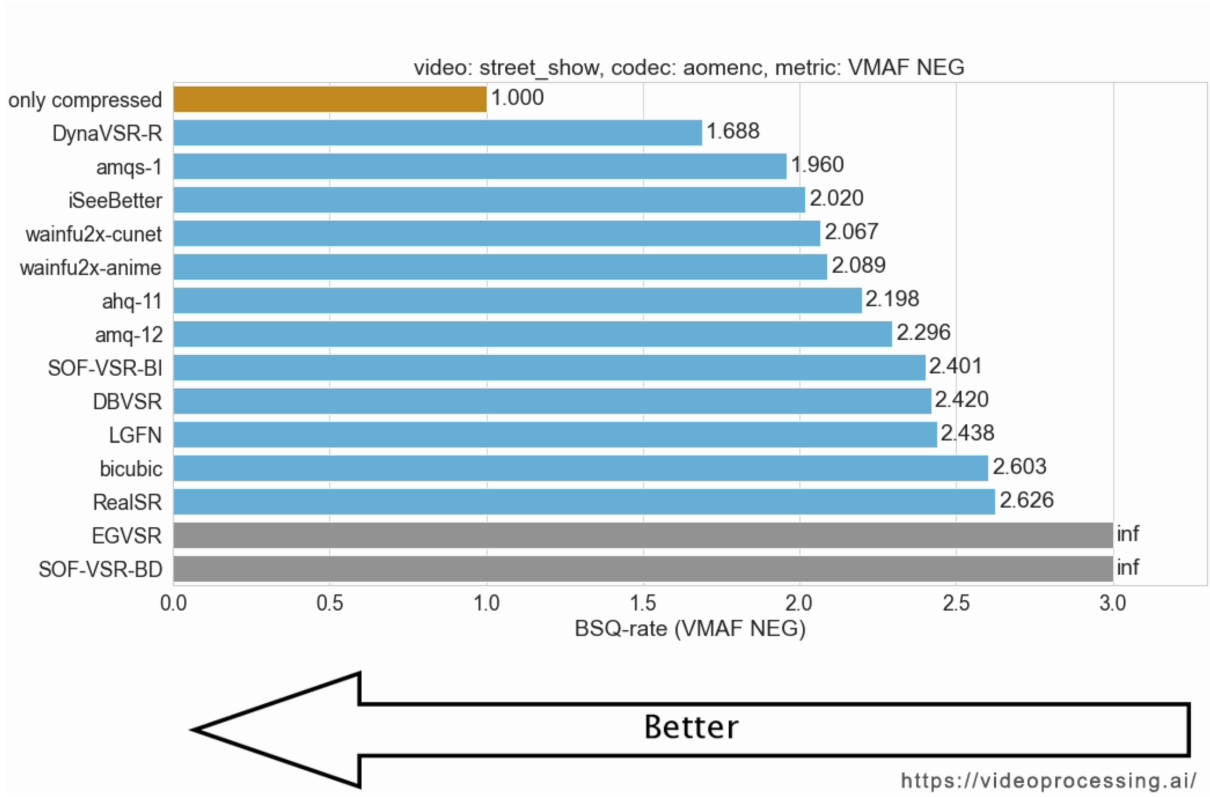


Figure 35b: BSQ-rate relative to “only compressed” —
street_show sequence, aomenc codec, VMAF NEG metric

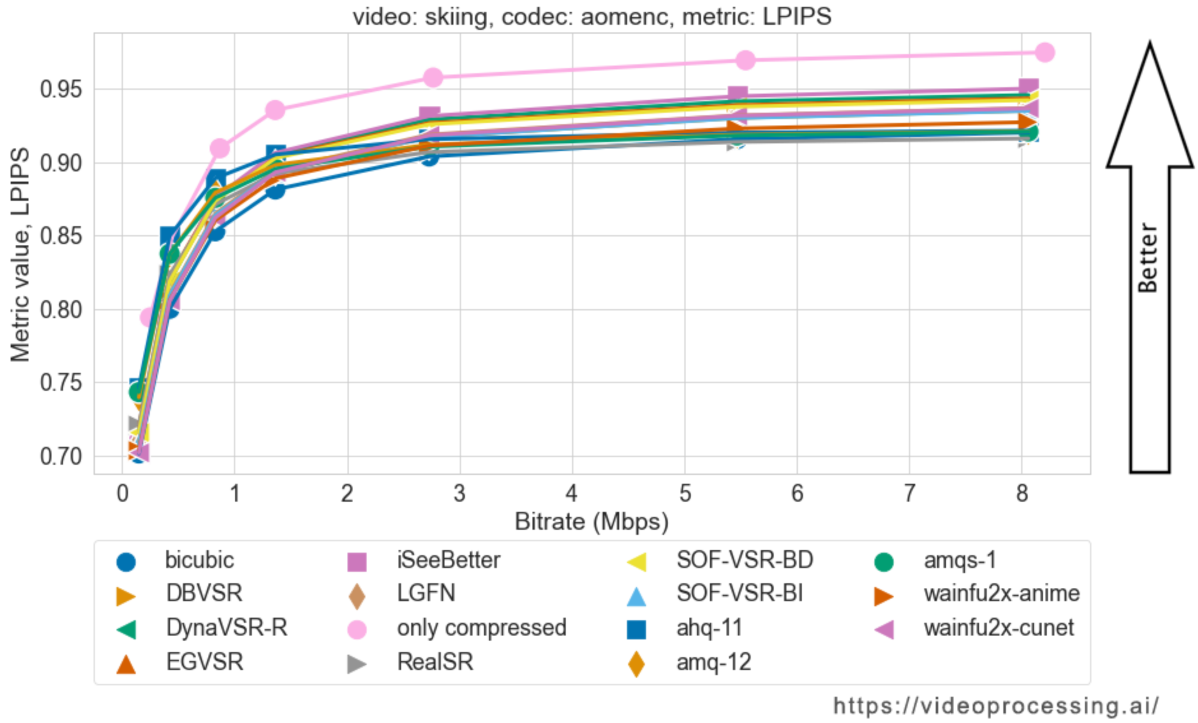


Figure 36a: Bitrate/Quality — *skiing* sequence, aomenc codec, LPIPS metric

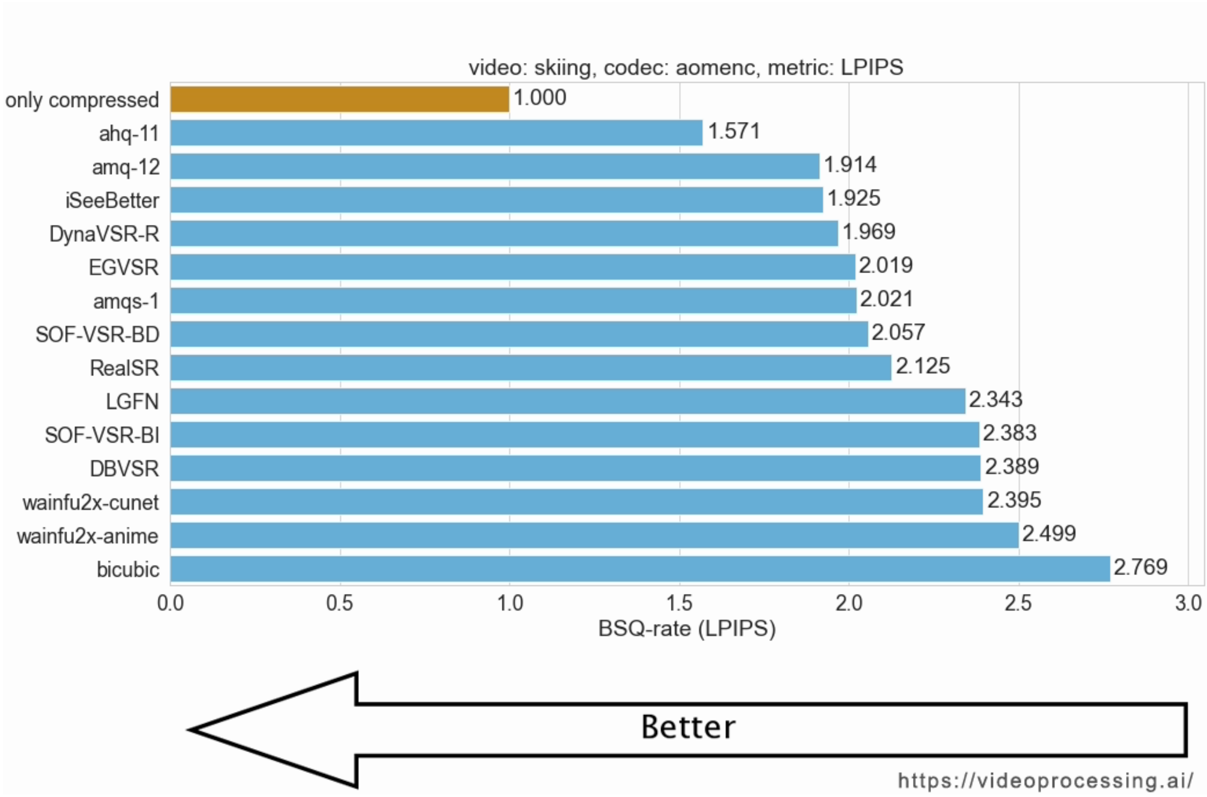


Figure 36b: BSQ-rate relative to “only compressed” — *skiing* sequence, aomenc codec, LPIPS metric

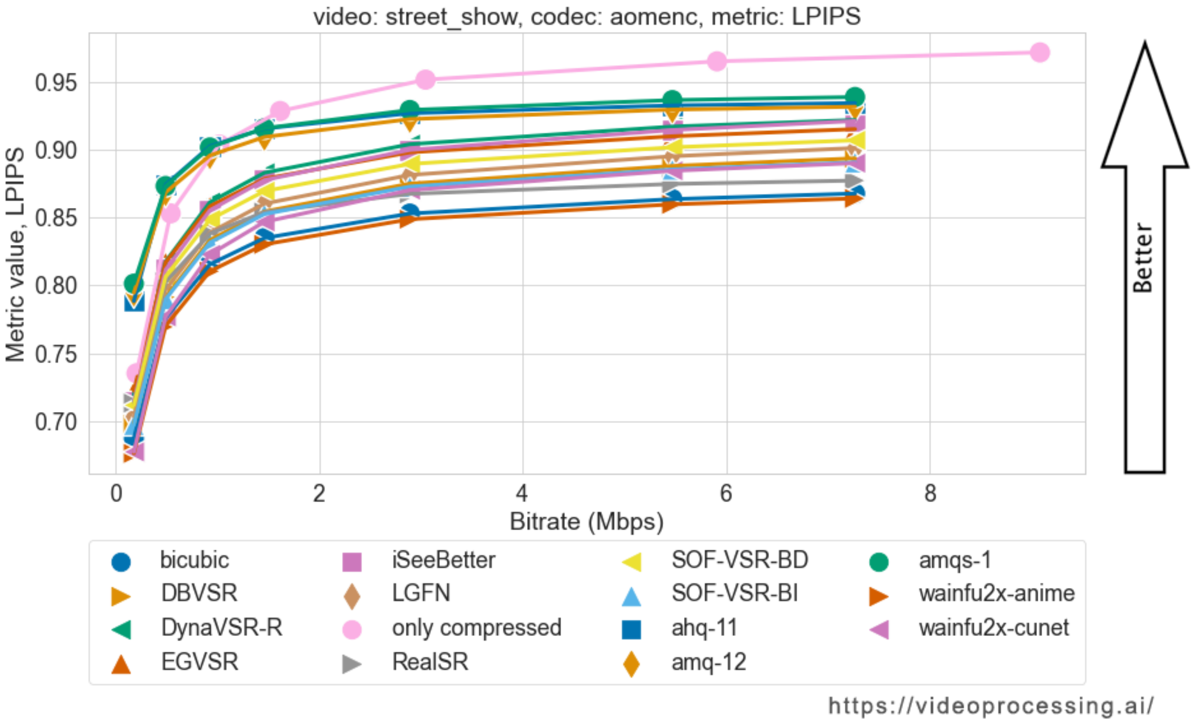


Figure 37a: Bitrate/Quality — *street_show* sequence, aomenc codec, LPIPS metric

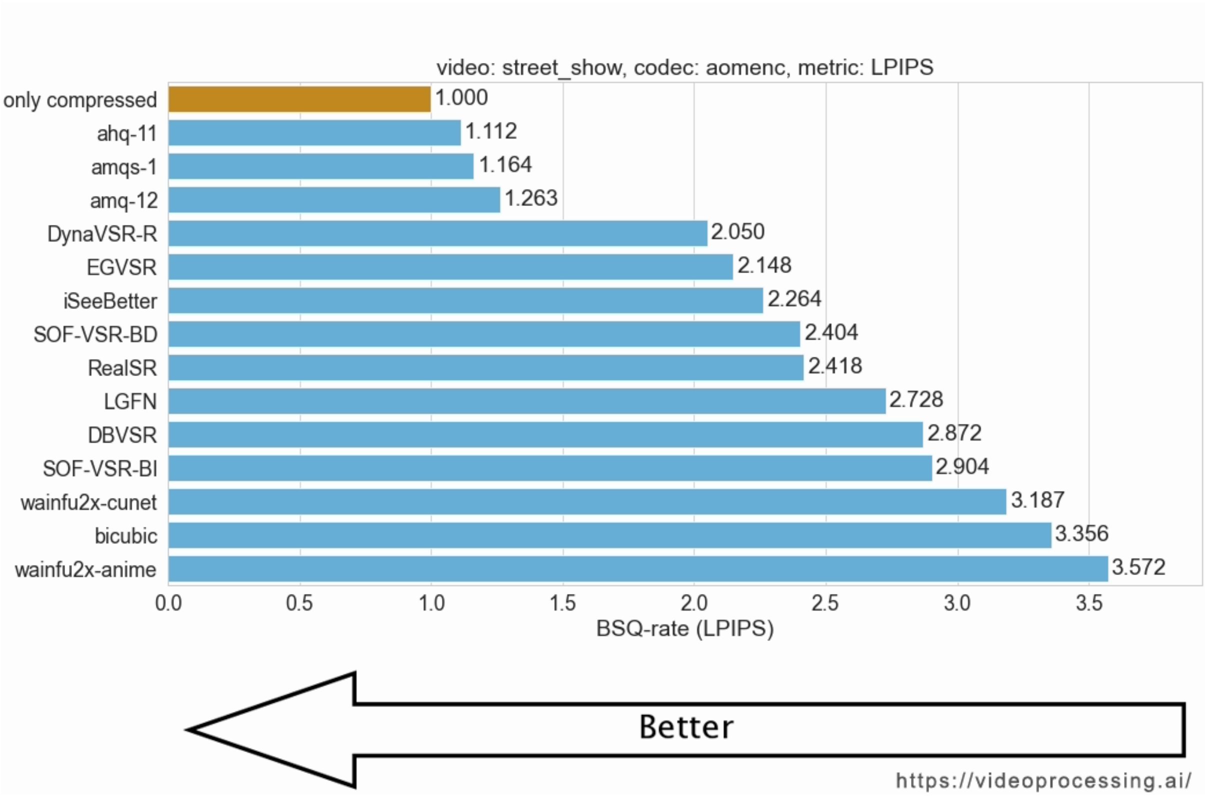


Figure 37b: BSQ-rate relative to “only compressed” —
street_show sequence, aomenc codec, LPIPS metric

In Figure 38 you can see the average BSQ-rate over each metric for the aomenc codec. “Only compressed” made by the x264 codec was used as a reference.

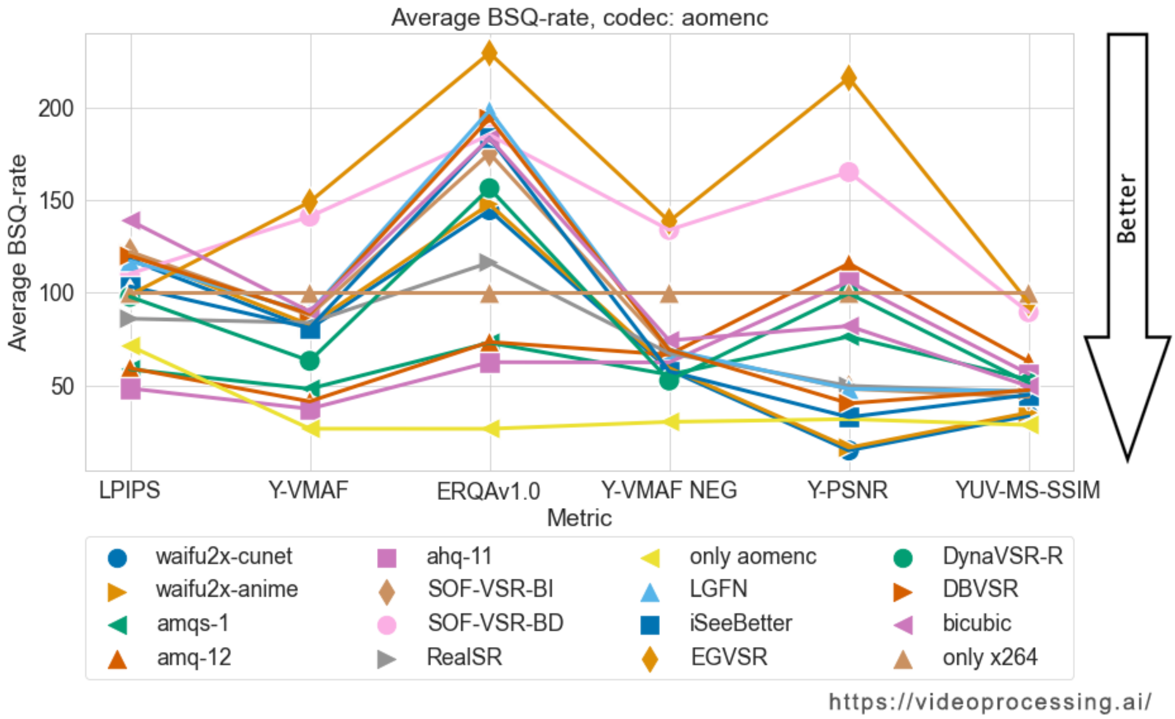


Figure 38: Average BSQ-rate relative to “only x264”.
SR input was compressed with the aomenc codec

2.5. VVenC results

In this section, you can see the results of applying SR models on videos compressed with the VVenC codec.

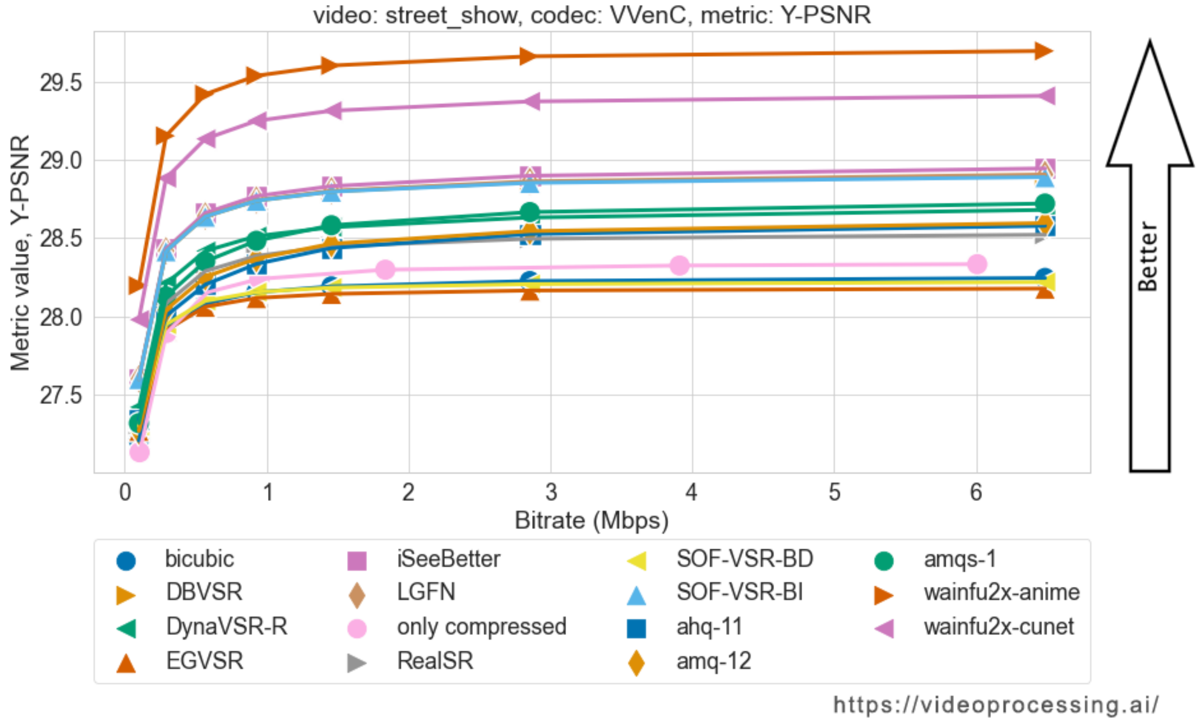


Figure 39a: Bitrate/Quality — *street_show* sequence, VVenC codec, Y-PSNR metric

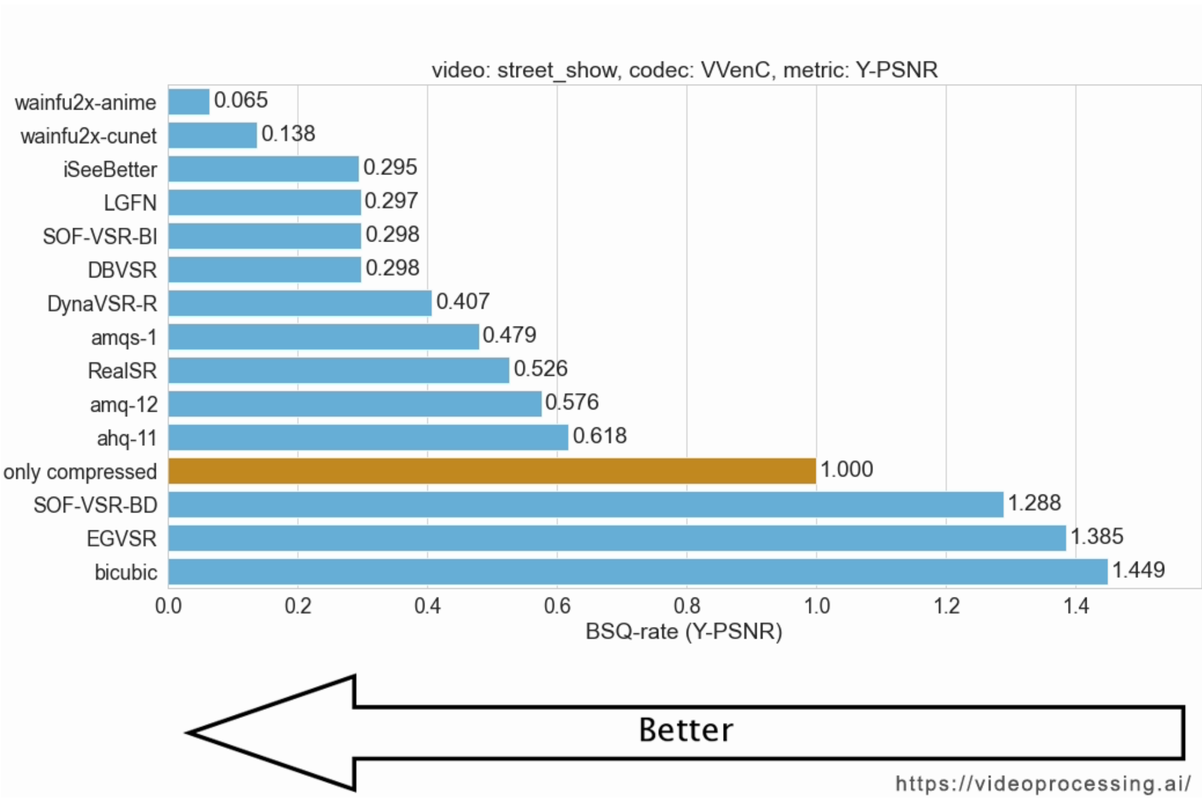


Figure 39b: BSQ-rate relative to “only compressed” — *street_show* sequence, VVenC codec, Y-PSNR metric

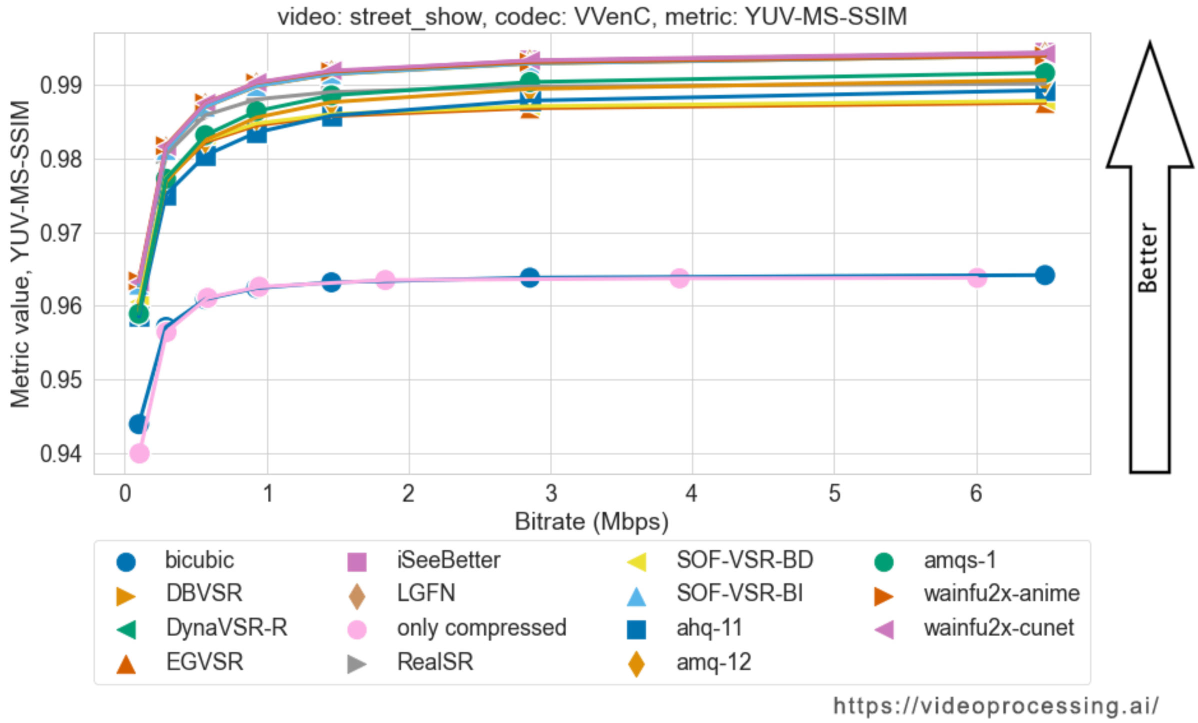


Figure 40a: Bitrate/Quality — *street_show* sequence, VVenC codec, YUV-MS-SSIM metric

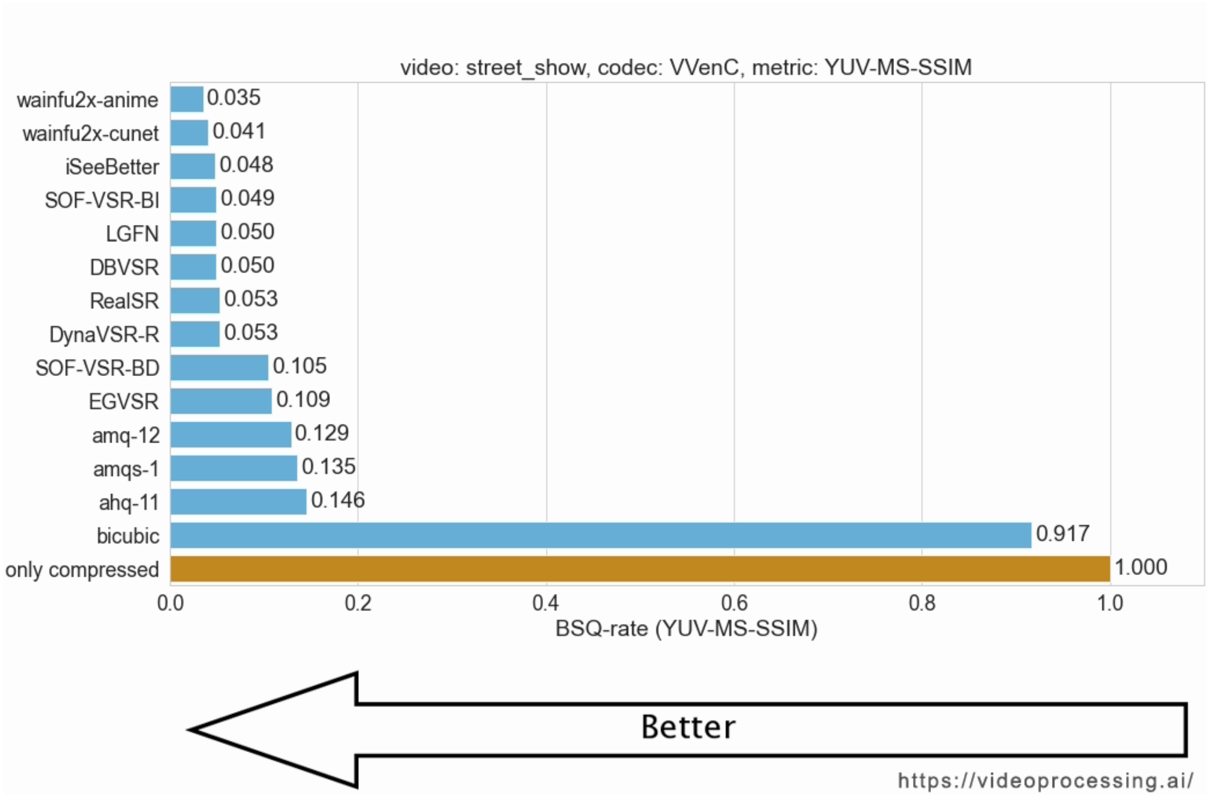


Figure 40b: BSQ-rate relative to “only compressed” —
street_show sequence, VVenC codec, YUV-MS-SSIM metric

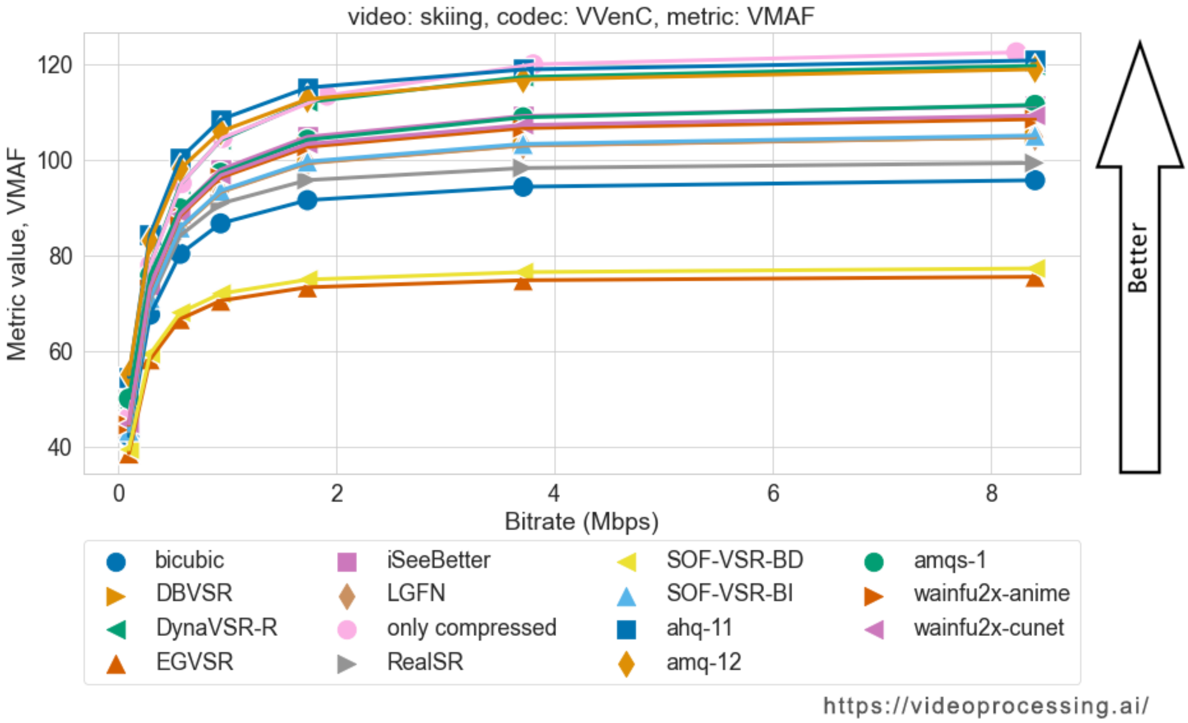


Figure 41a: Bitrate/Quality — *skiing* sequence, VVenC codec, VMAF metric

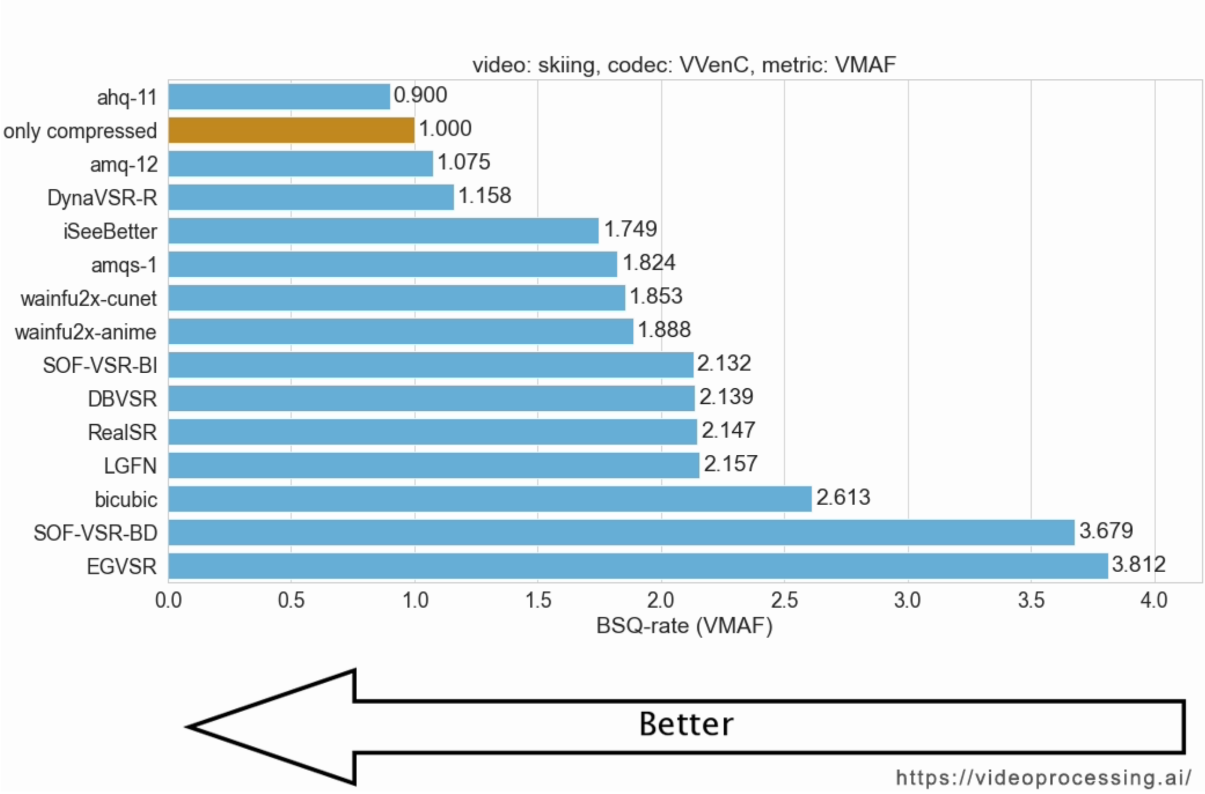


Figure 41b: BSQ-rate relative to “only compressed” —
skiing sequence, VVenC codec, VMAF metric

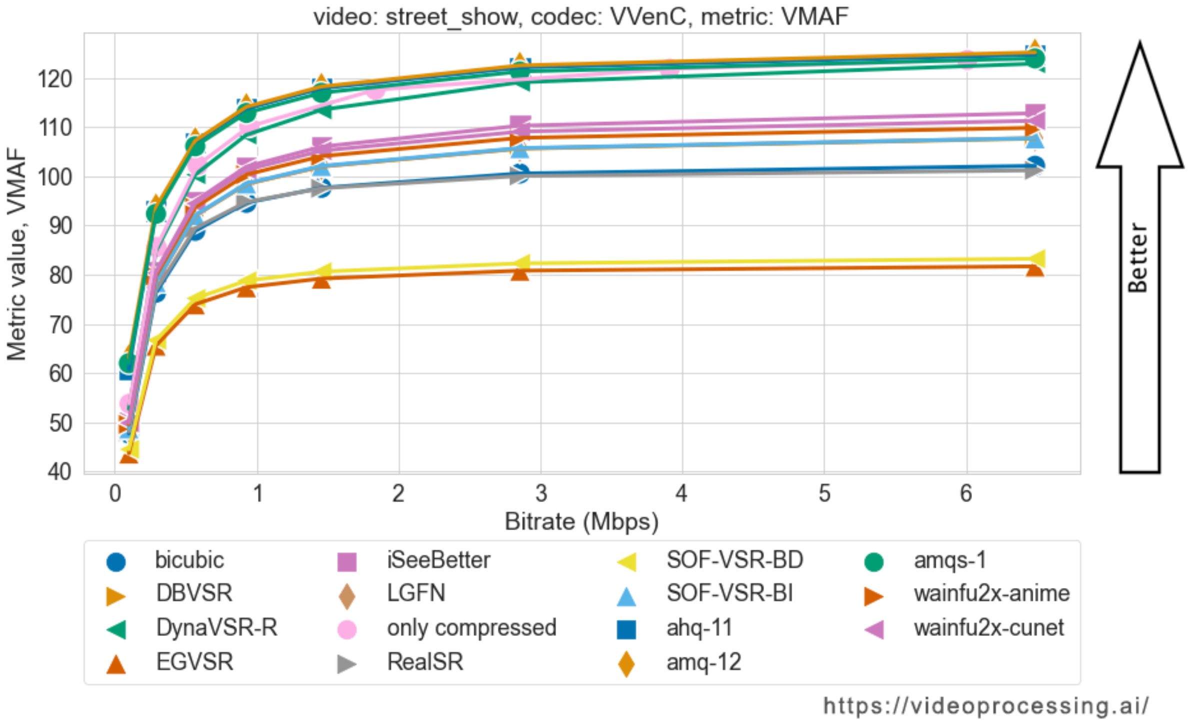


Figure 42a: Bitrate/Quality — *street_show* sequence, VVenC codec, VMAF metric

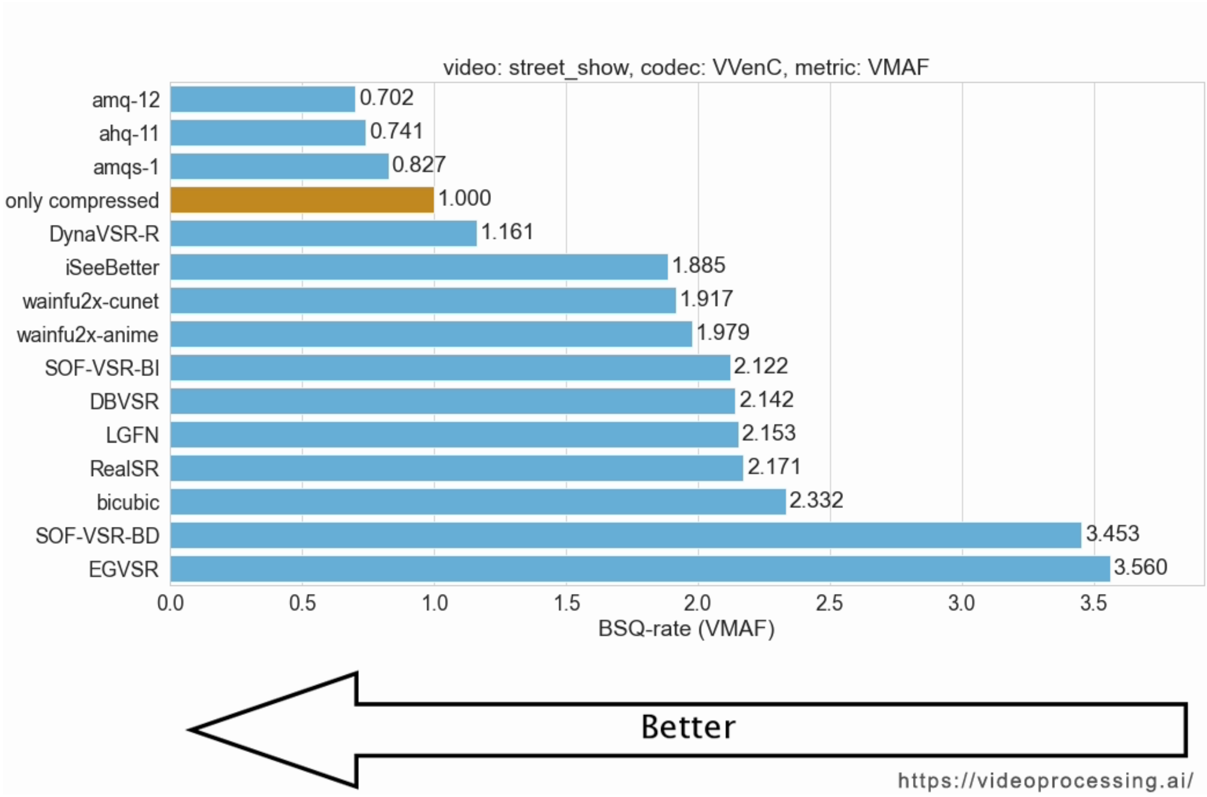


Figure 42b: BSQ-rate relative to "only compressed" —
street_show sequence, VVenC codec, VMAF metric

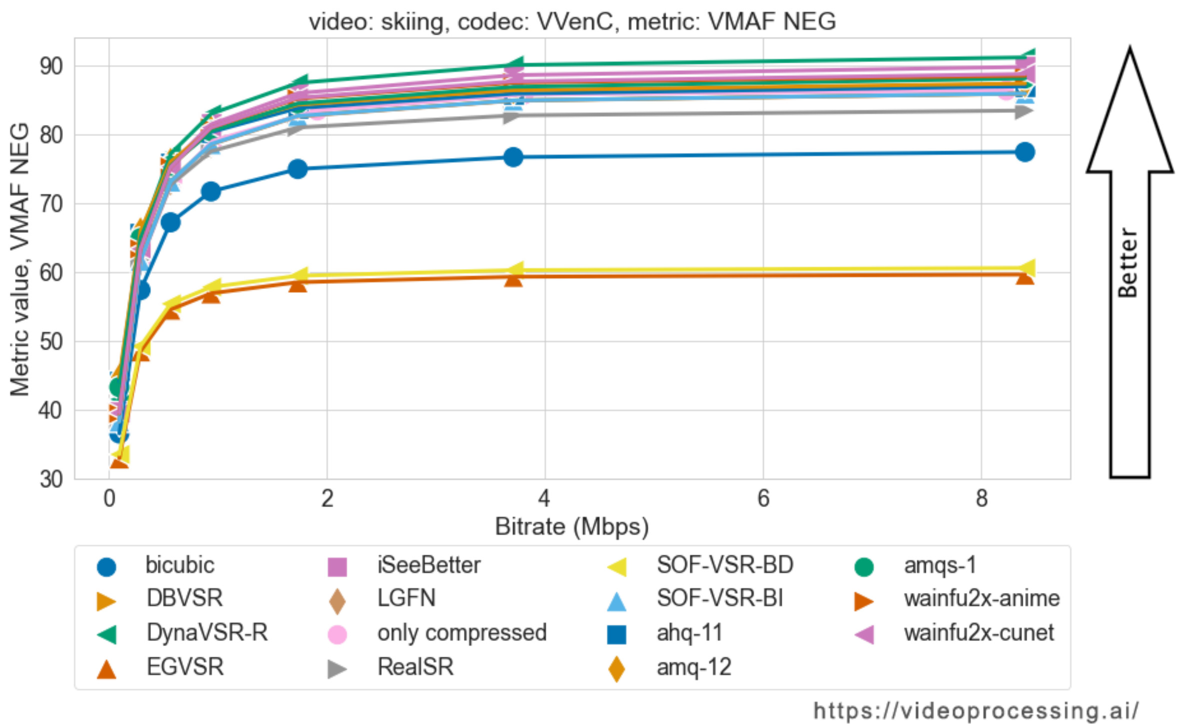


Figure 43a: Bitrate/Quality — *skiing* sequence, VVenC codec, VMAF NEG metric

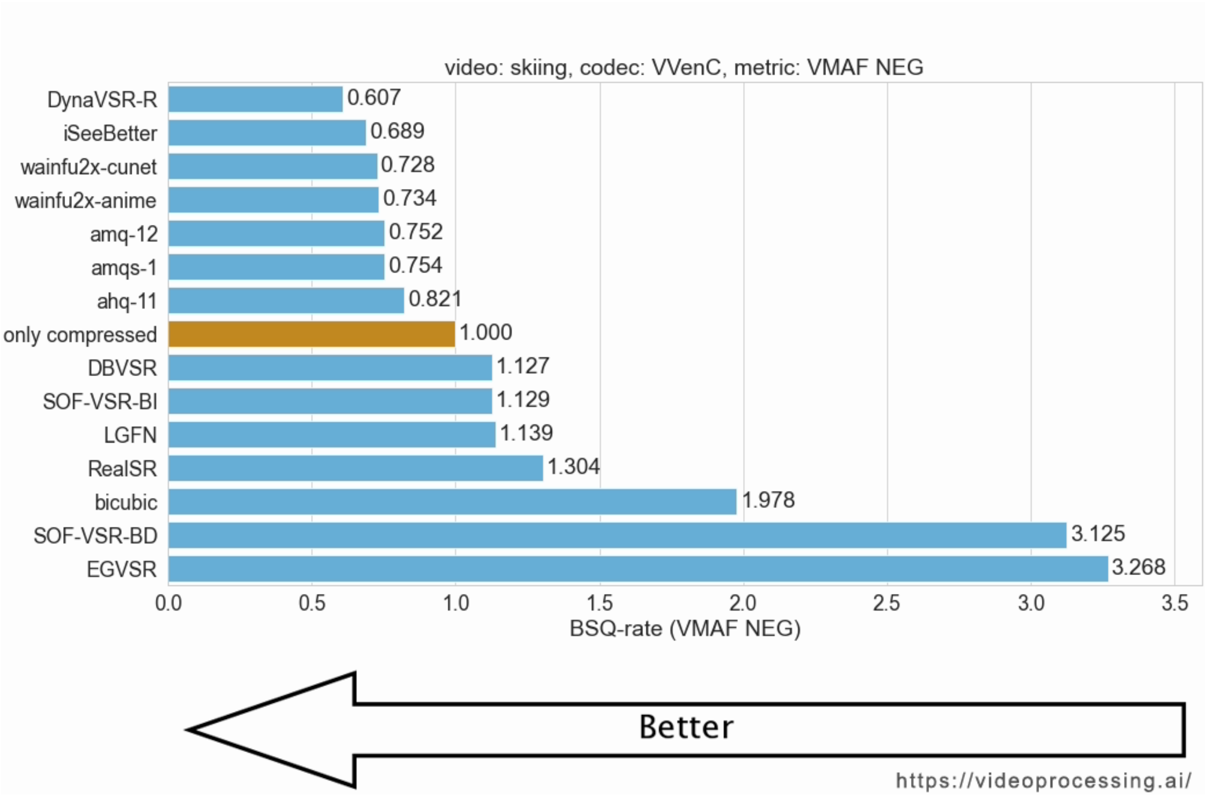


Figure 43b: BSQ-rate relative to “only compressed” — *skiing* sequence, VVenC codec, VMAF NEG metric

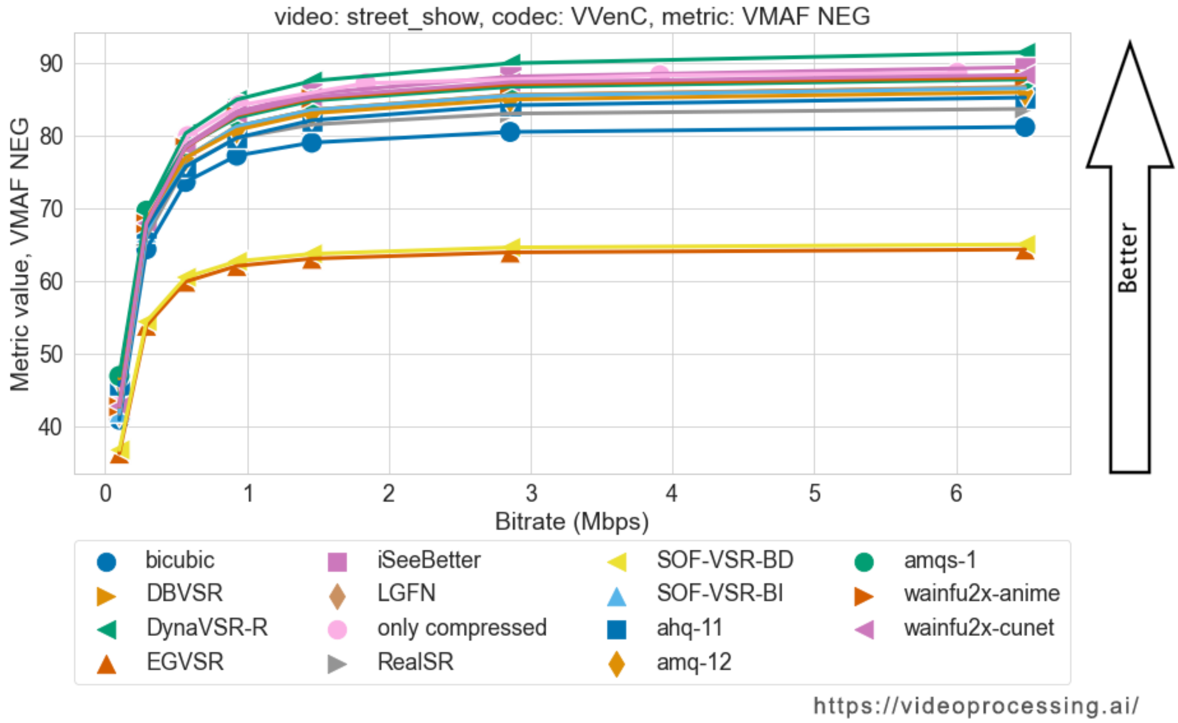


Figure 44a: Bitrate/Quality — *street_show* sequence, VVenC codec, VMAF NEG metric

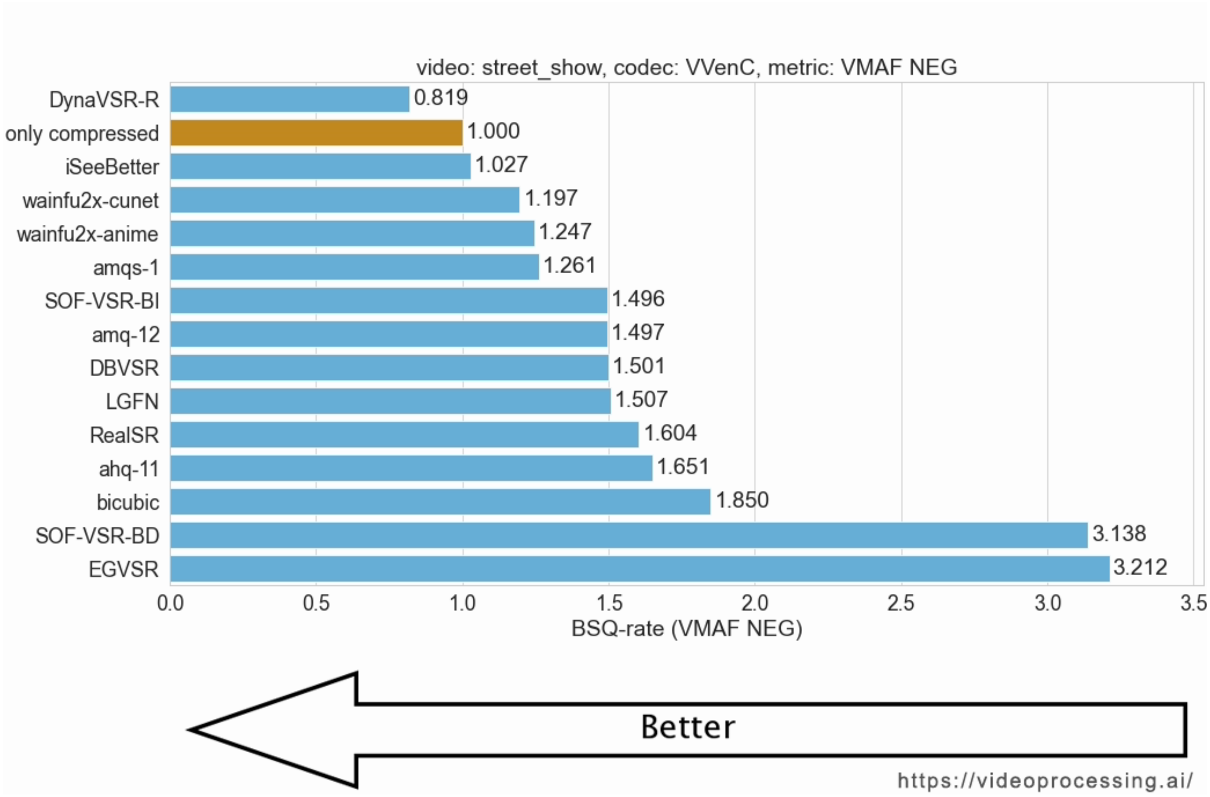


Figure 44b: BSQ-rate relative to “only compressed” —
street_show sequence, VVenC codec, VMAF NEG metric

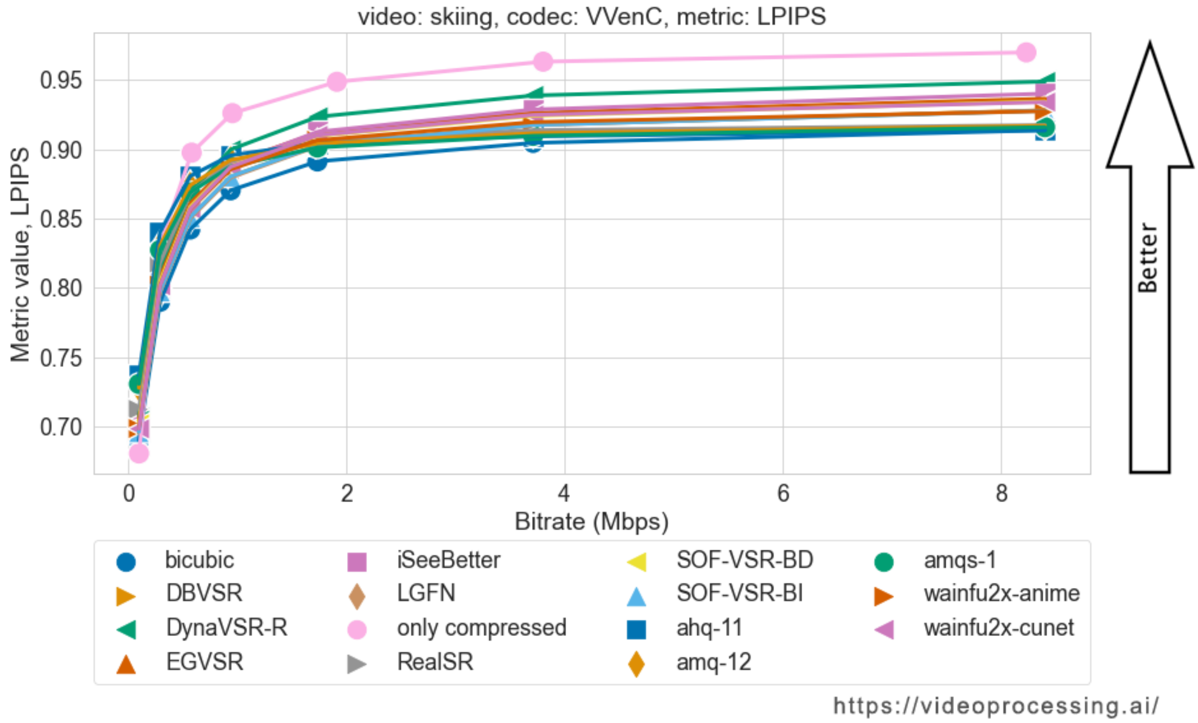


Figure 45a: Bitrate/Quality — *skiing* sequence, VVenC codec, LPIPS metric

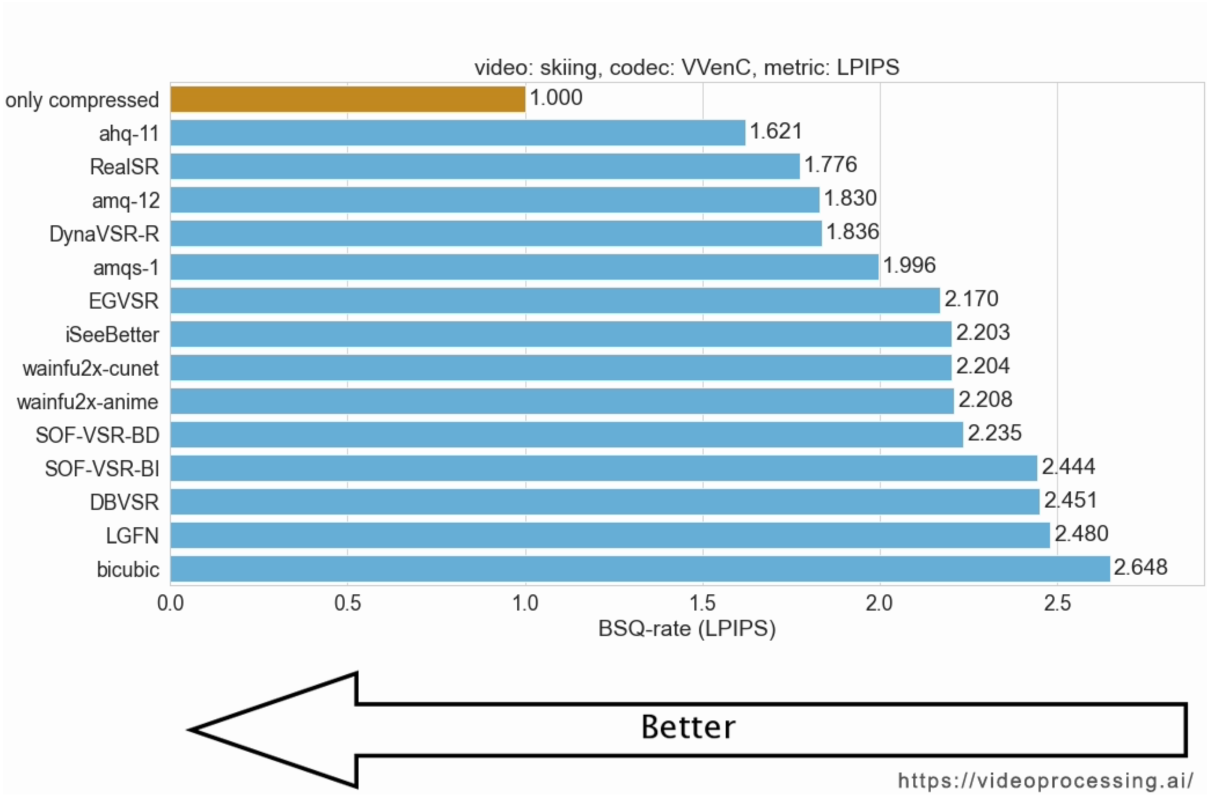


Figure 45b: BSQ-rate relative to “only compressed” —
skiing sequence, VVenC codec, LPIPS metric

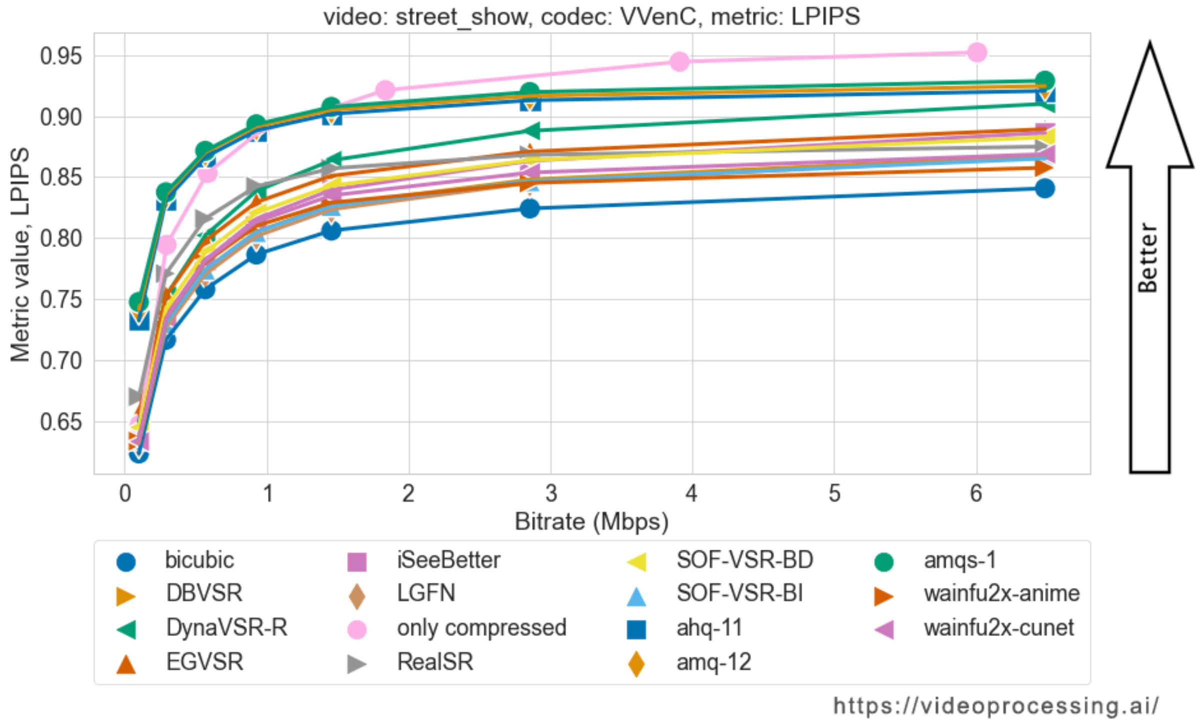


Figure 46a: Bitrate/Quality — *street_show* sequence, VVenC codec, LPIPS metric

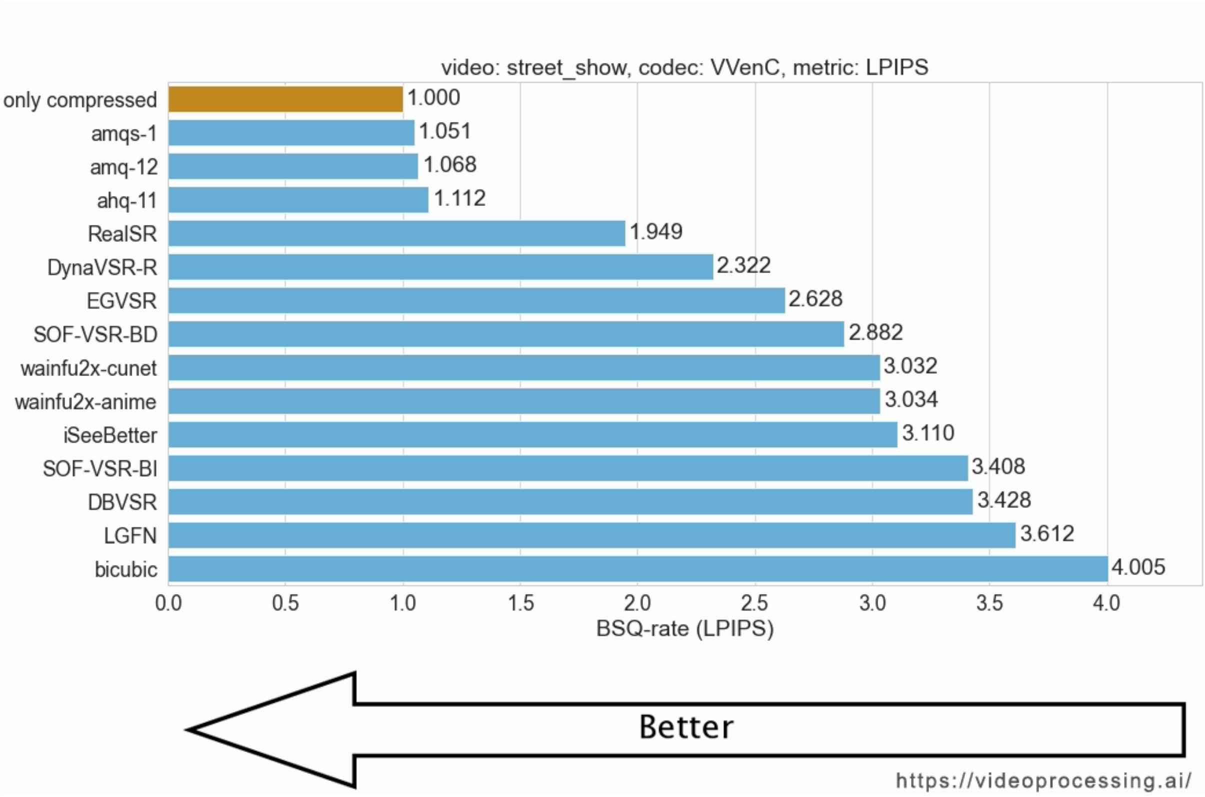


Figure 46b: BSQ-rate relative to “only compressed” —
street_show sequence, VVenC codec, LPIPS metric

In Figure 47 you can see the average BSQ-rate over each metric for the VVenC codec. “Only compressed” made by x264 codec was used as a reference.

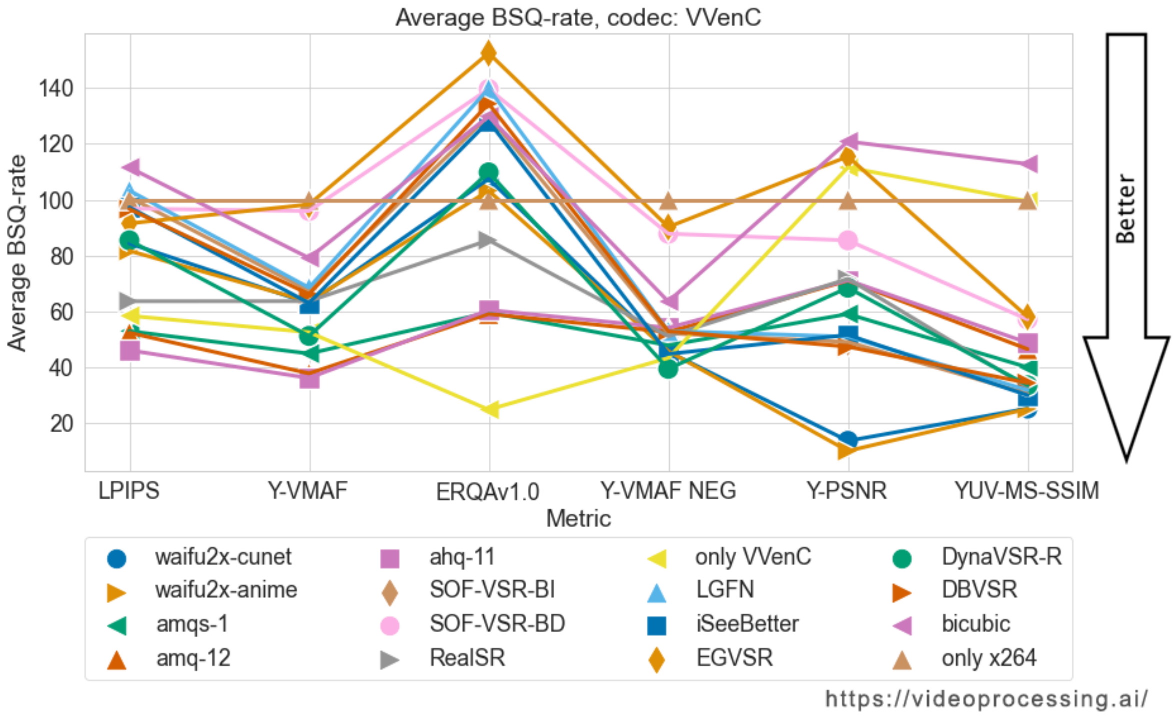


Figure 47: Average BSQ-rate relative to “only x264”.
SR input was compressed with the VVenC codec

2.6. uavs3e results

In this section, you can see the results of applying SR models on videos compressed with the uavs3e codec.

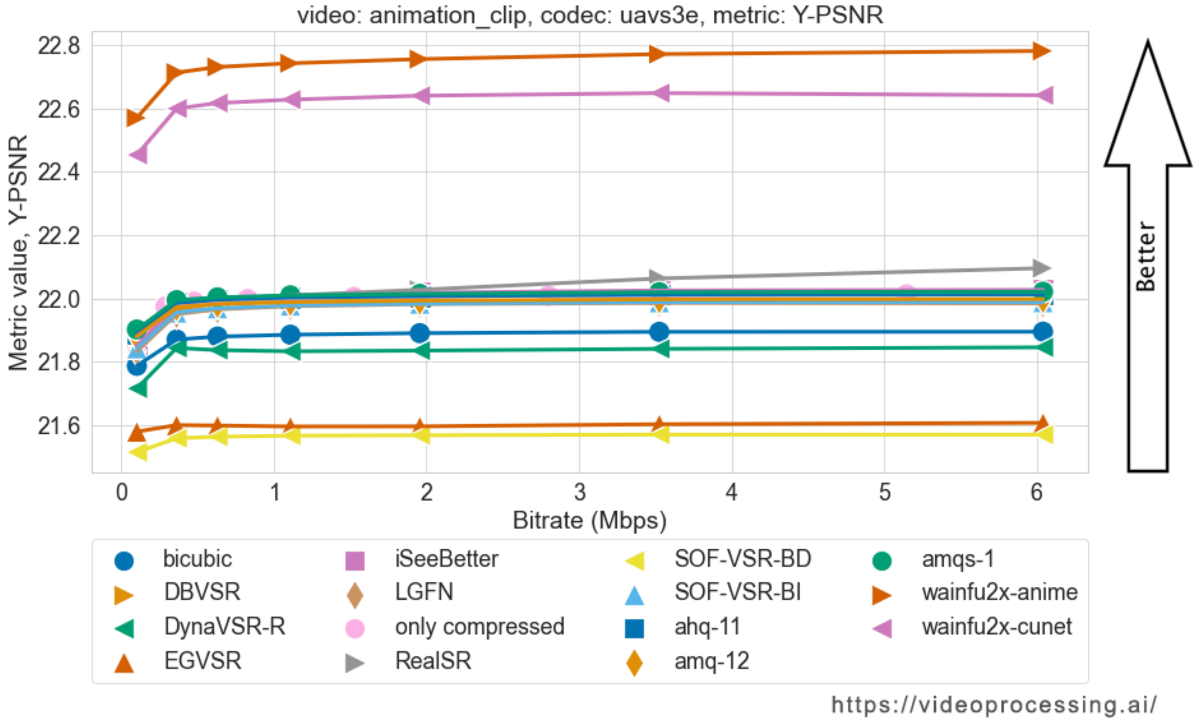


Figure 48a: Bitrate/Quality — *animation_clip* sequence, uavs3e codec, Y-PSNR metric

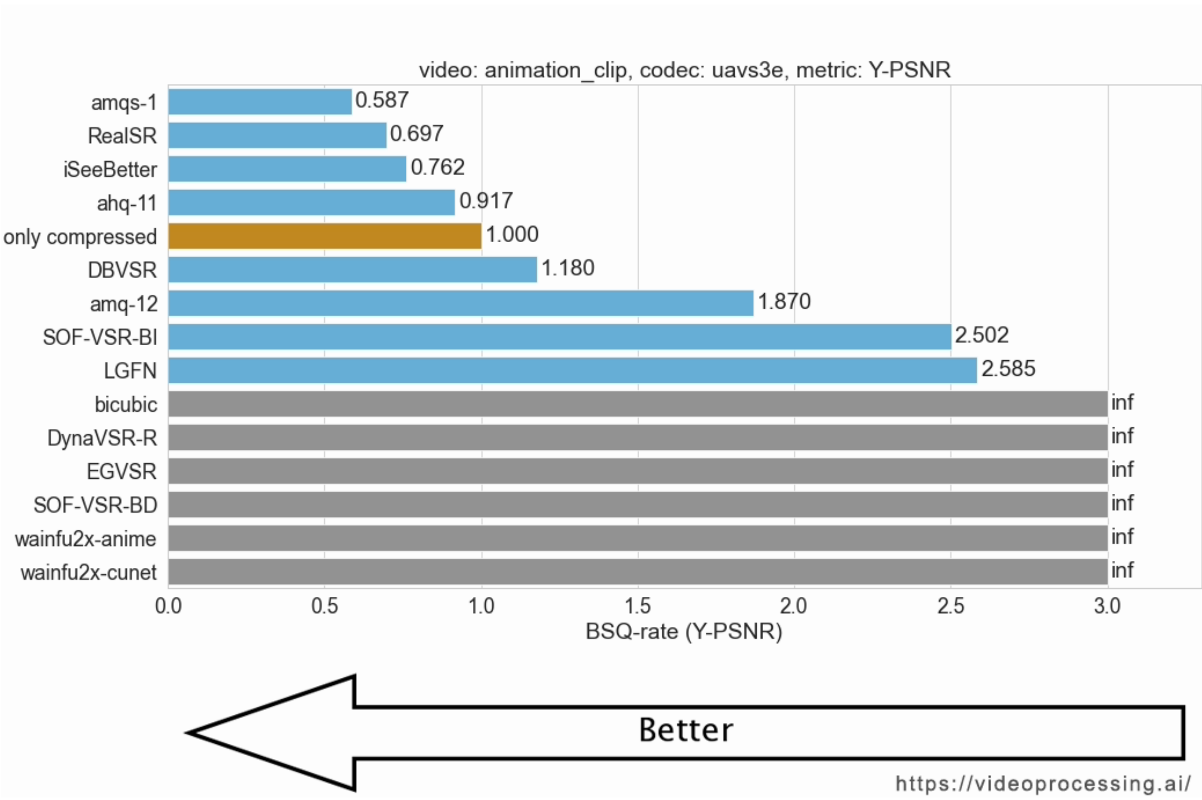


Figure 48b: BSQ-rate relative to “only compressed” — *animation_clip* sequence, uavs3e codec, Y-PSNR metric

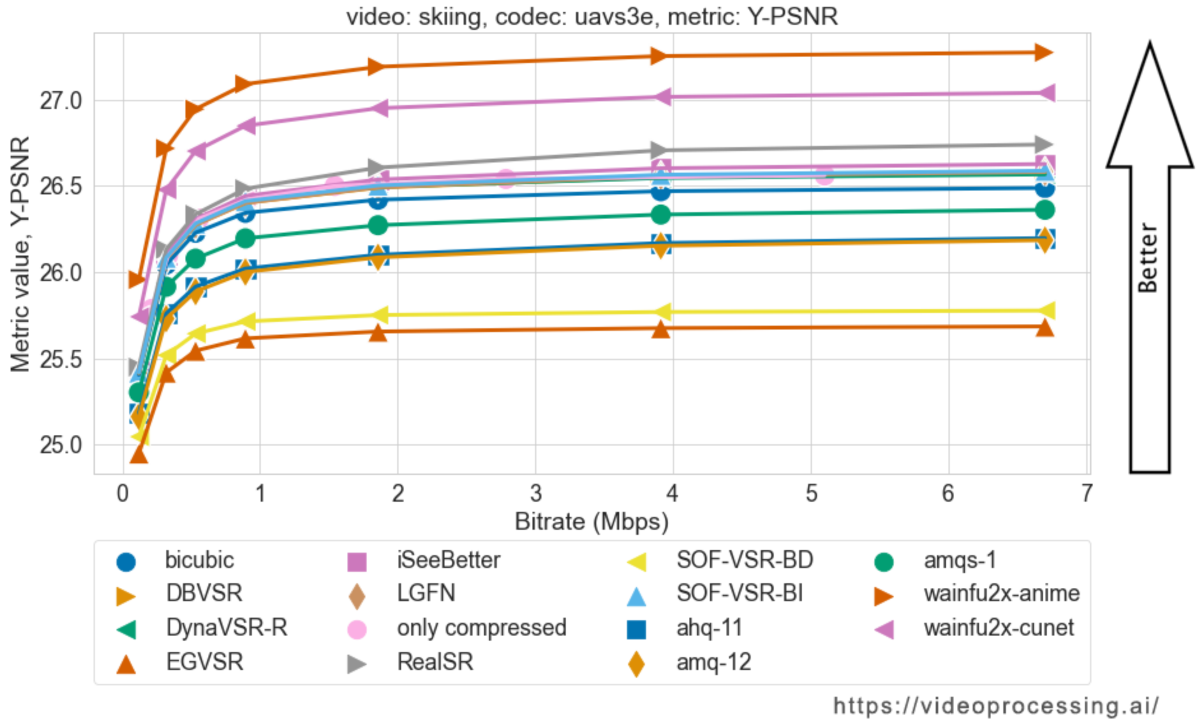


Figure 49a: Bitrate/Quality — *skiing* sequence, uavs3e codec, Y-PSNR metric

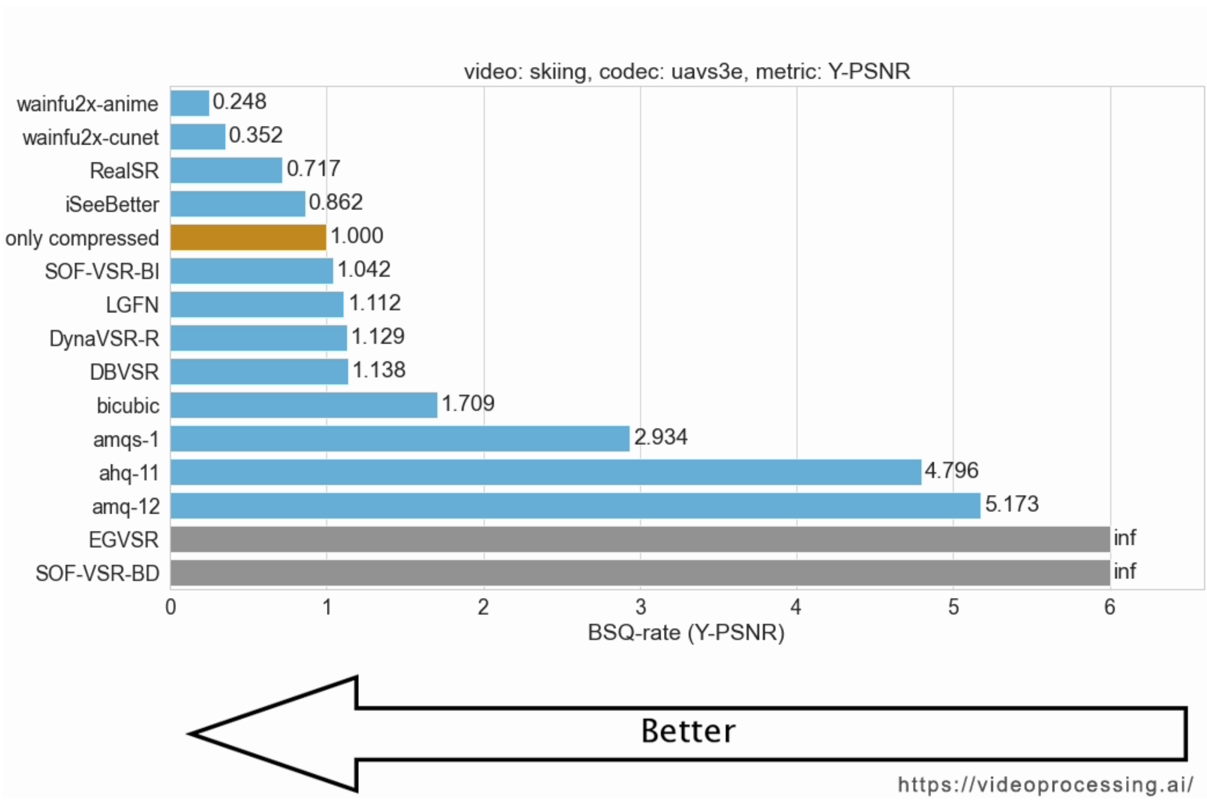


Figure 49b: BSQ-rate relative to “only compressed” —
skiing sequence, uavs3e codec, Y-PSNR metric

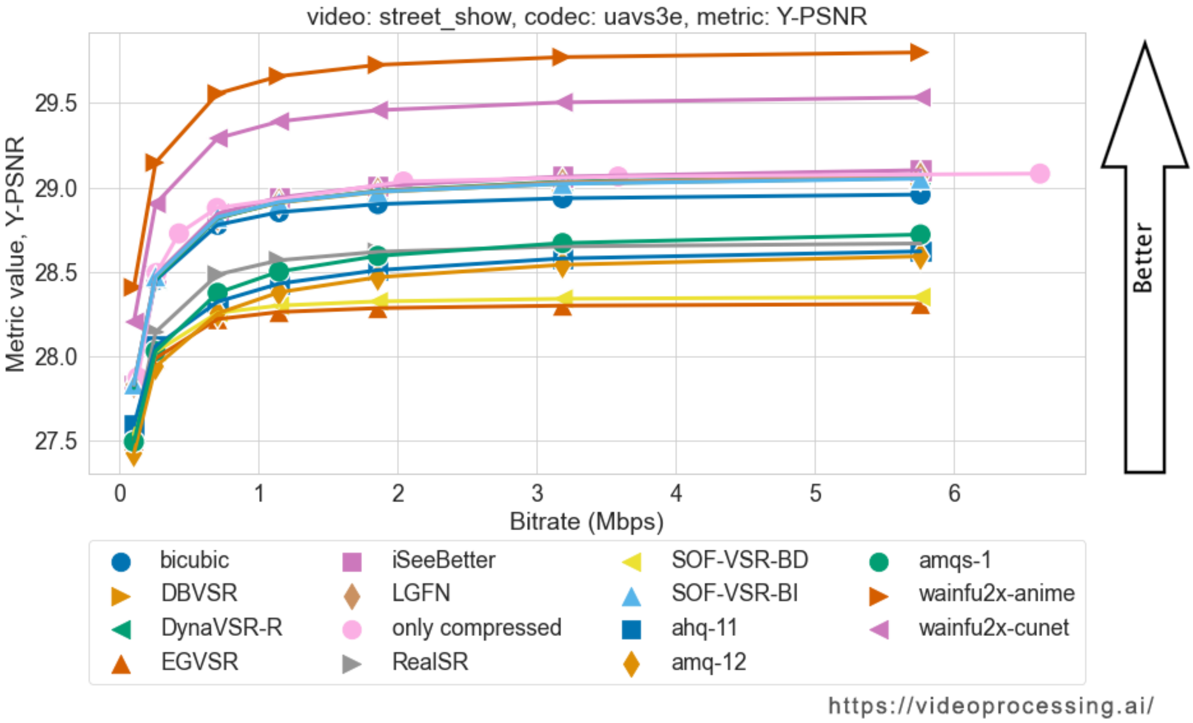


Figure 50a: Bitrate/Quality — *street_show* sequence, uavs3e codec, Y-PSNR metric

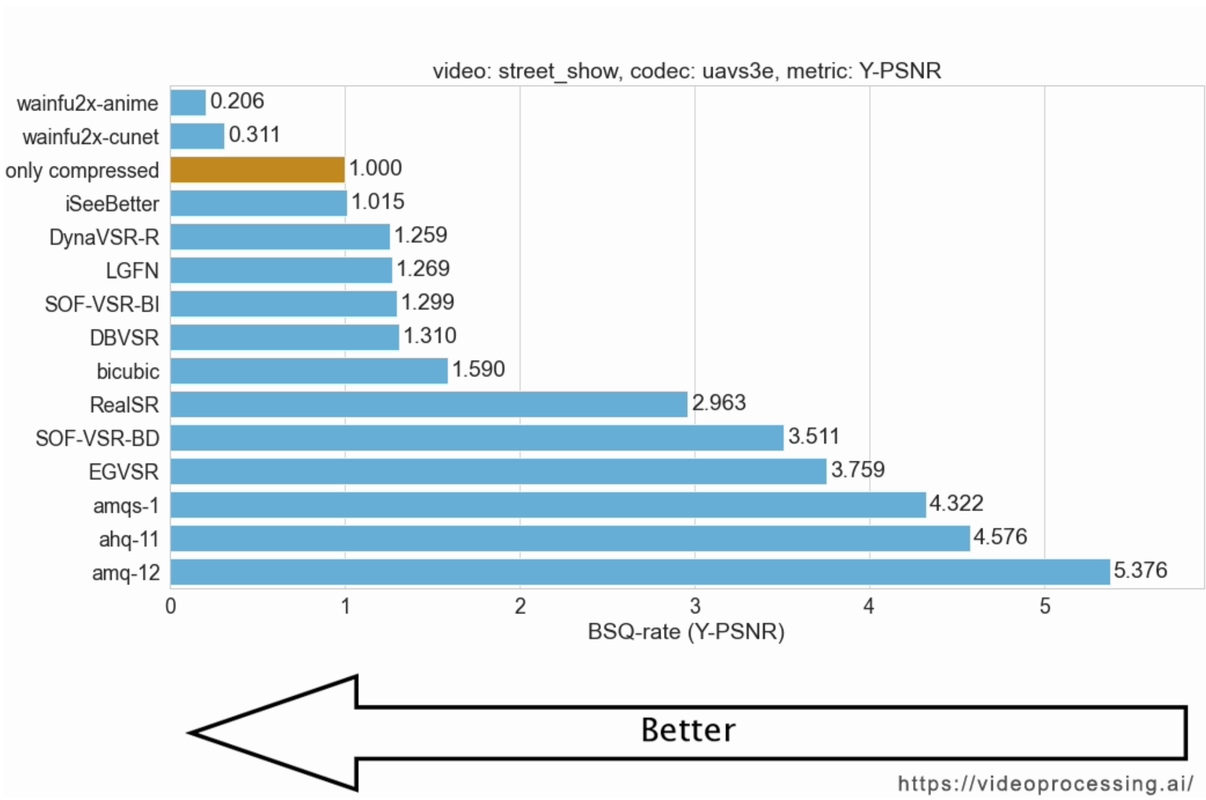


Figure 50b: BSQ-rate relative to “only compressed” — *street_show* sequence, uavs3e codec, Y-PSNR metric

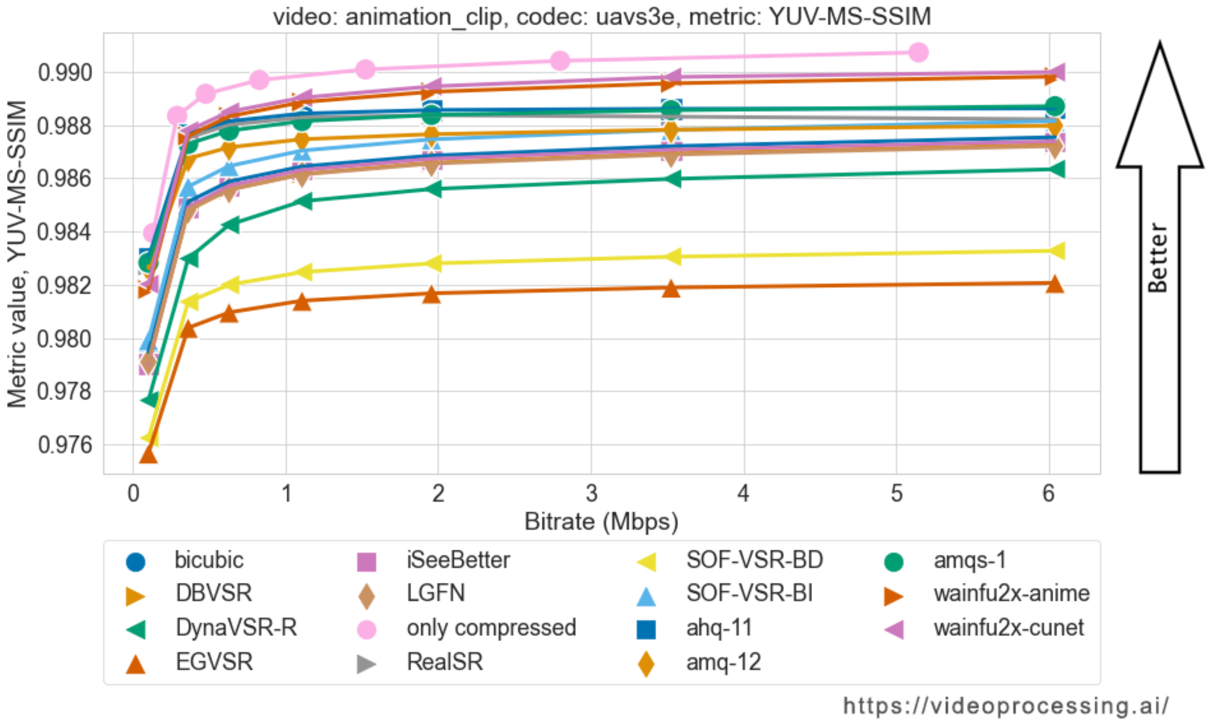


Figure 51a: Bitrate/Quality — *animation_clip* sequence, uavs3e codec, YUV-MS-SSIM metric

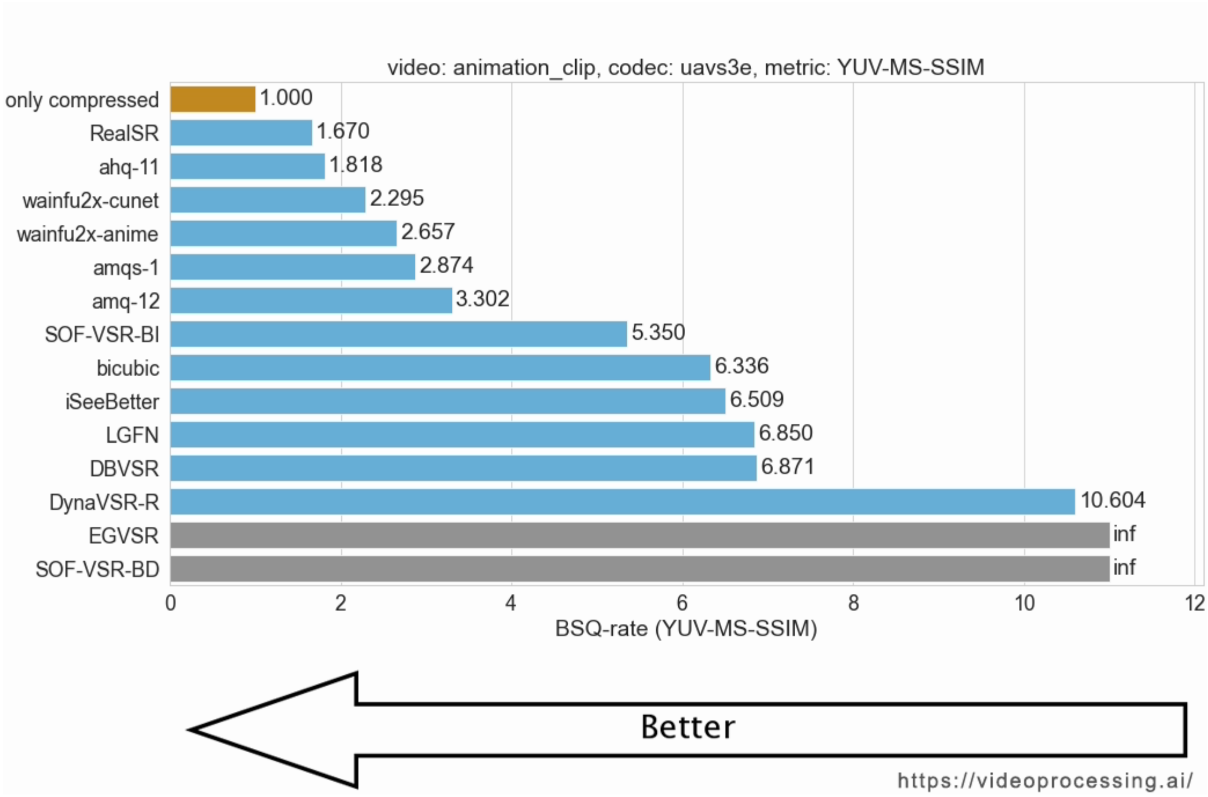


Figure 51b: BSQ-rate relative to “only compressed” —
animation_clip sequence, uavs3e codec, YUV-MS-SSIM metric

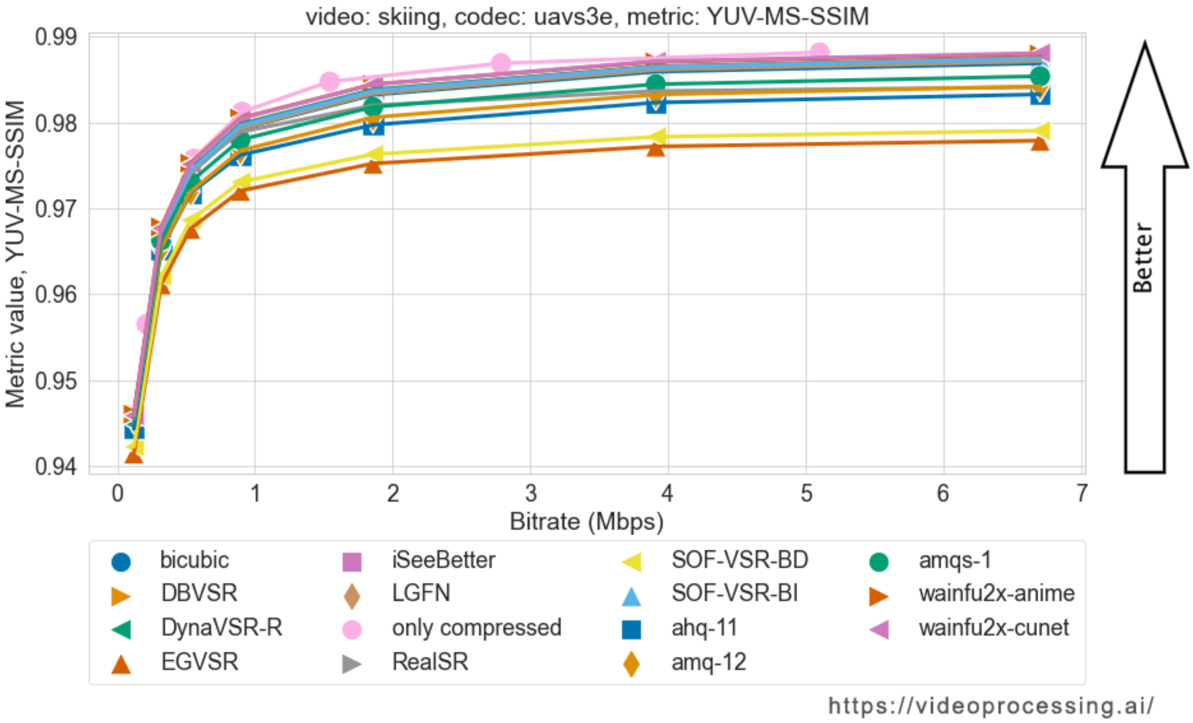


Figure 52a: Bitrate/Quality — *skiing* sequence, uavs3e codec, YUV-MS-SSIM metric

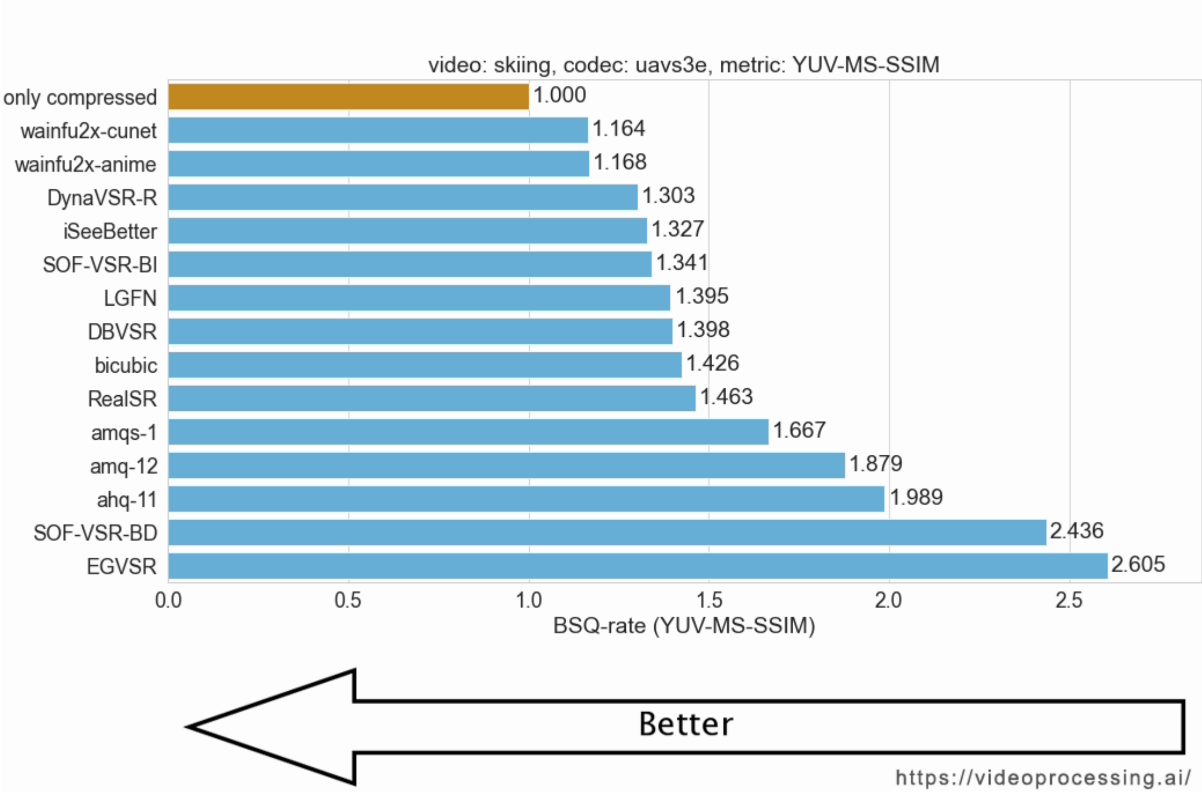


Figure 52b: BSQ-rate relative to “only compressed” —
skiing sequence, uavs3e codec, YUV-MS-SSIM metric

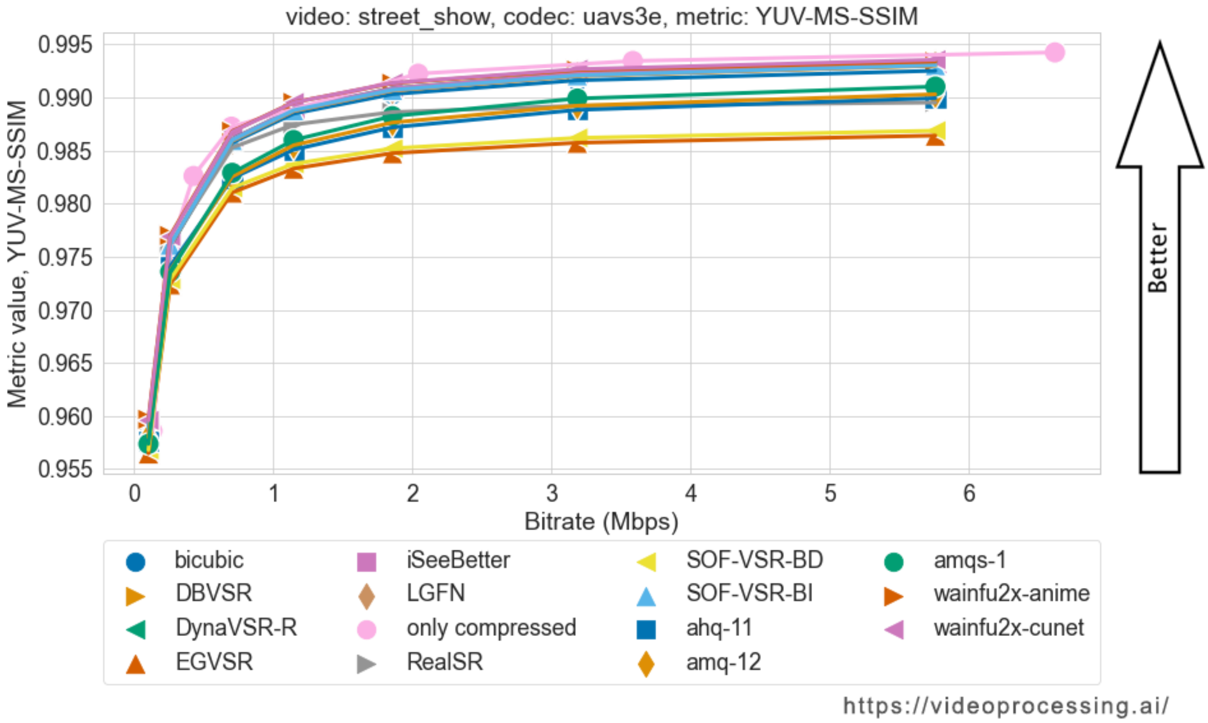


Figure 53a: Bitrate/Quality — *street_show* sequence, uavs3e codec, YUV-MS-SSIM metric

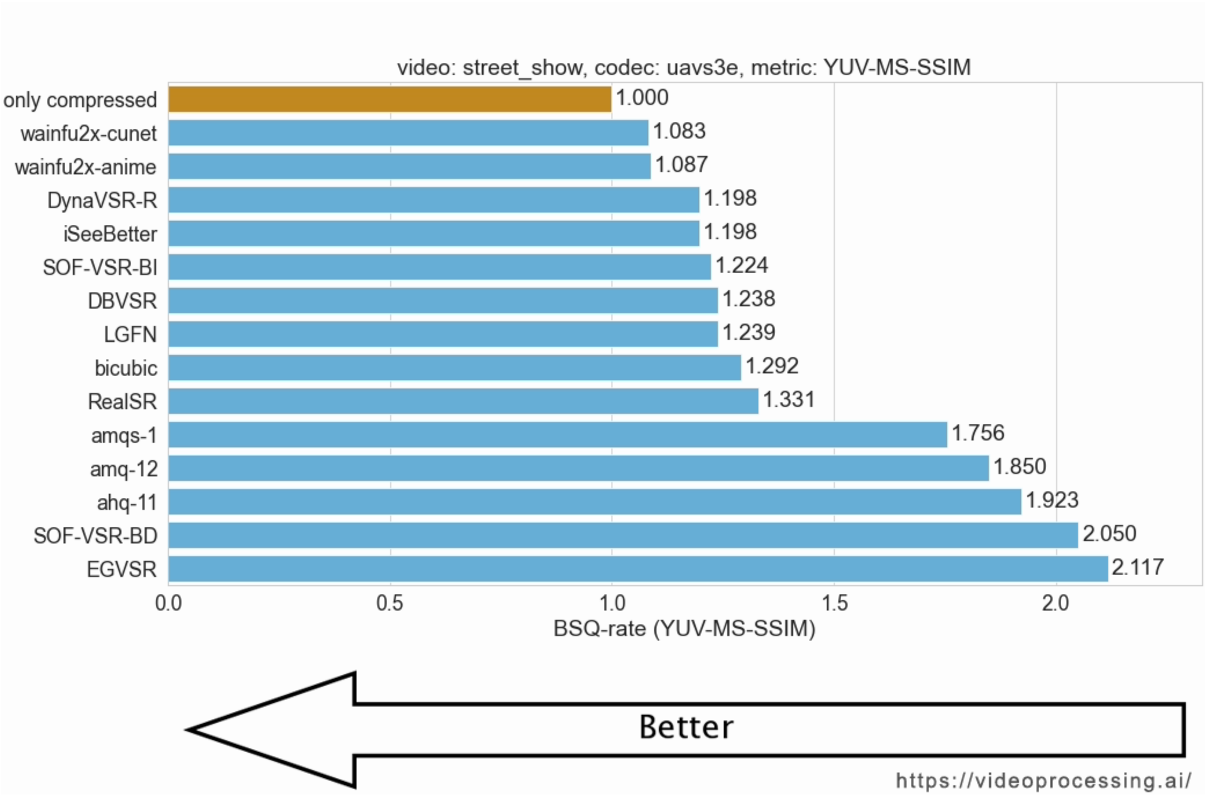


Figure 53b: BSQ-rate relative to “only compressed” —
street_show sequence, uavs3e codec, YUV-MS-SSIM metric

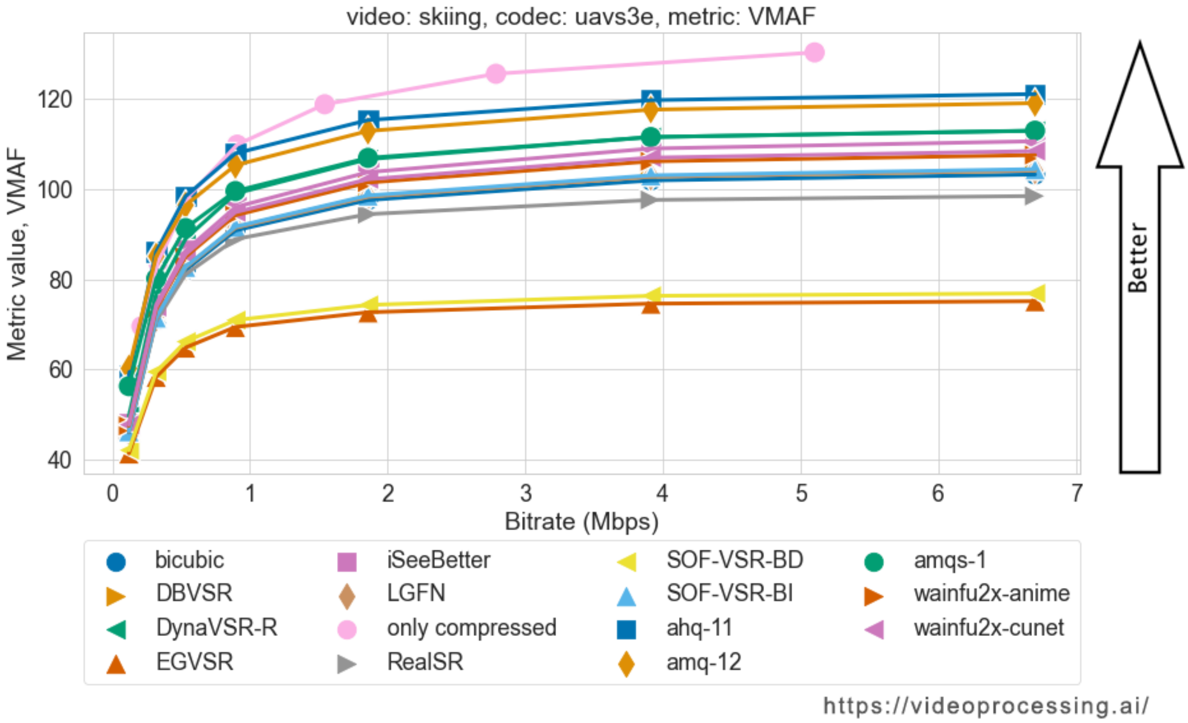


Figure 54a: Bitrate/Quality — *skiing* sequence, uavs3e codec, VMAF metric

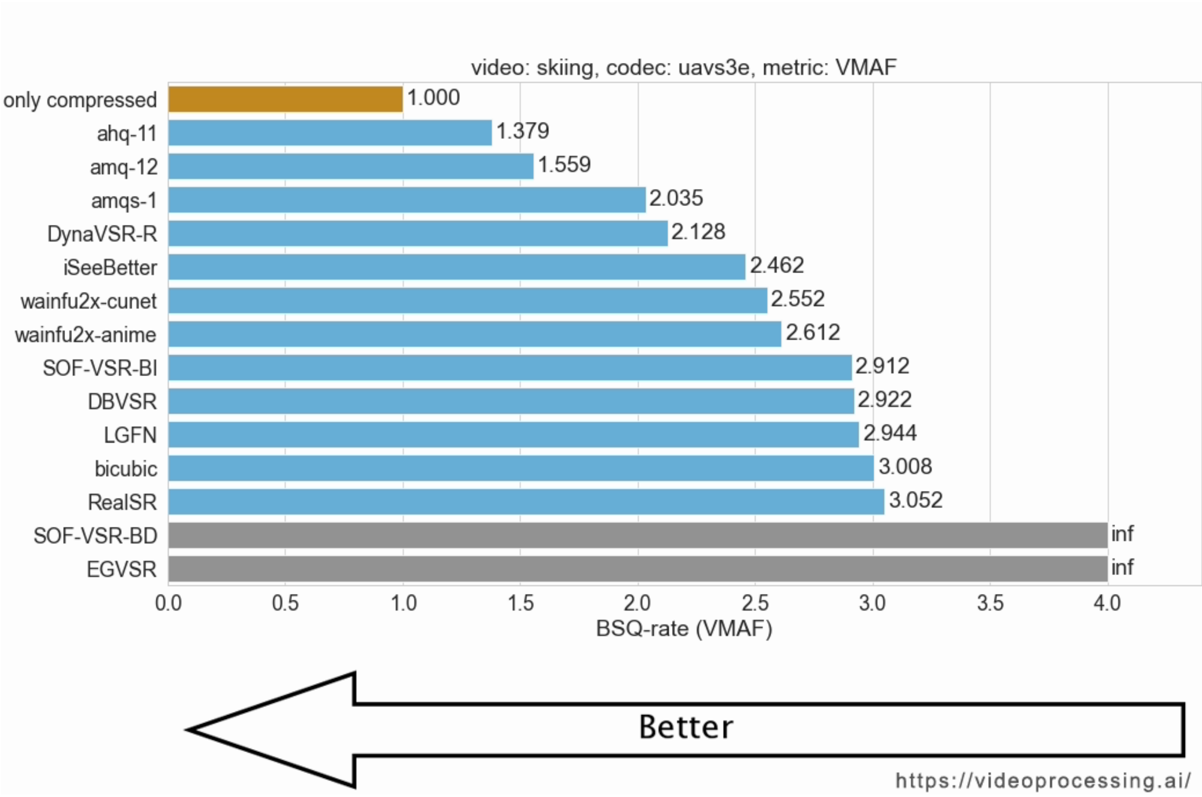


Figure 54b: BSQ-rate relative to “only compressed” —
skiing sequence, uavs3e codec, VMAF metric

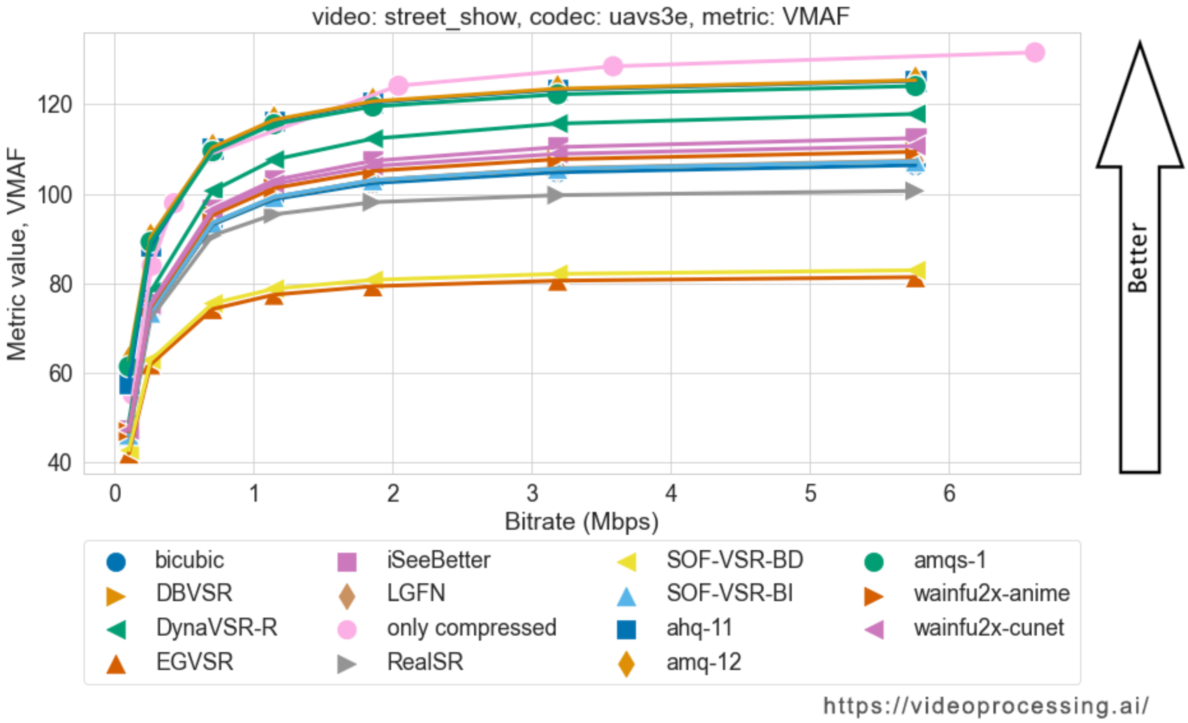


Figure 55a: Bitrate/Quality — *street_show* sequence, uavs3e codec, VMAF metric

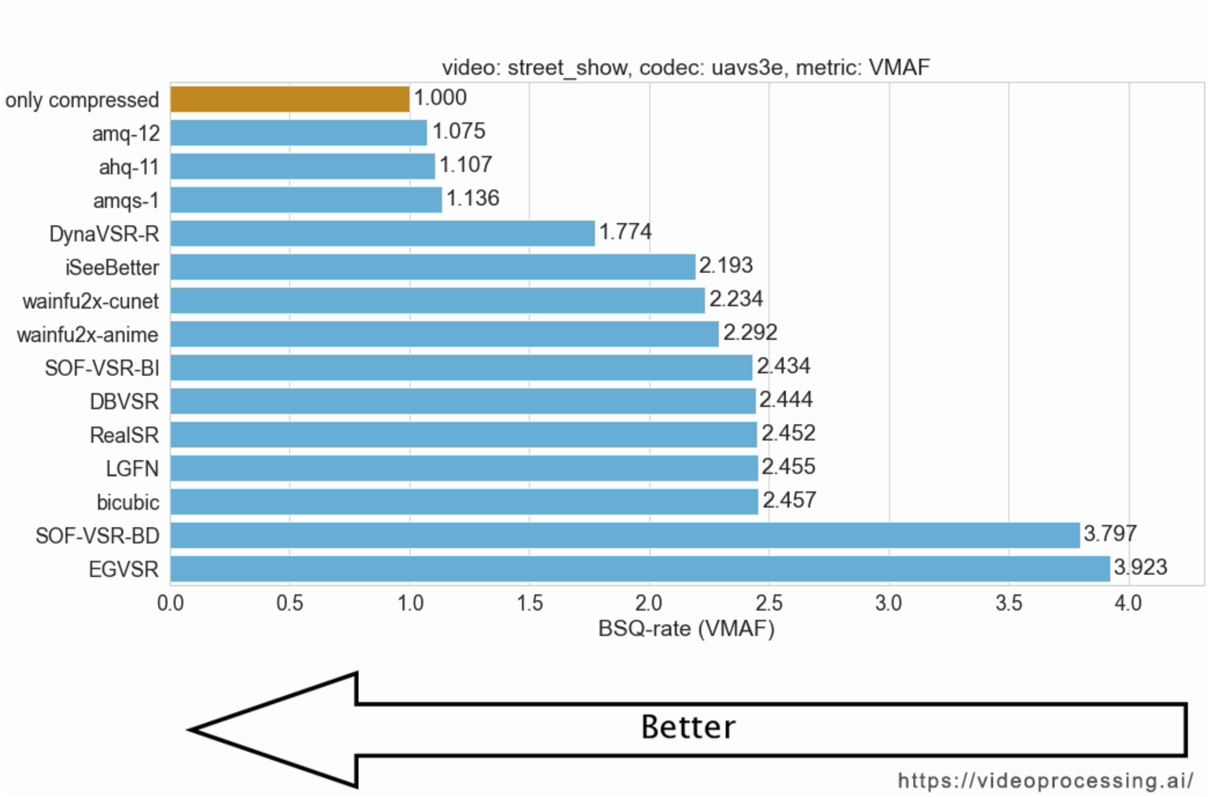


Figure 55b: BSQ-rate relative to “only compressed” —
street_show sequence, uavs3e codec, VMAF metric

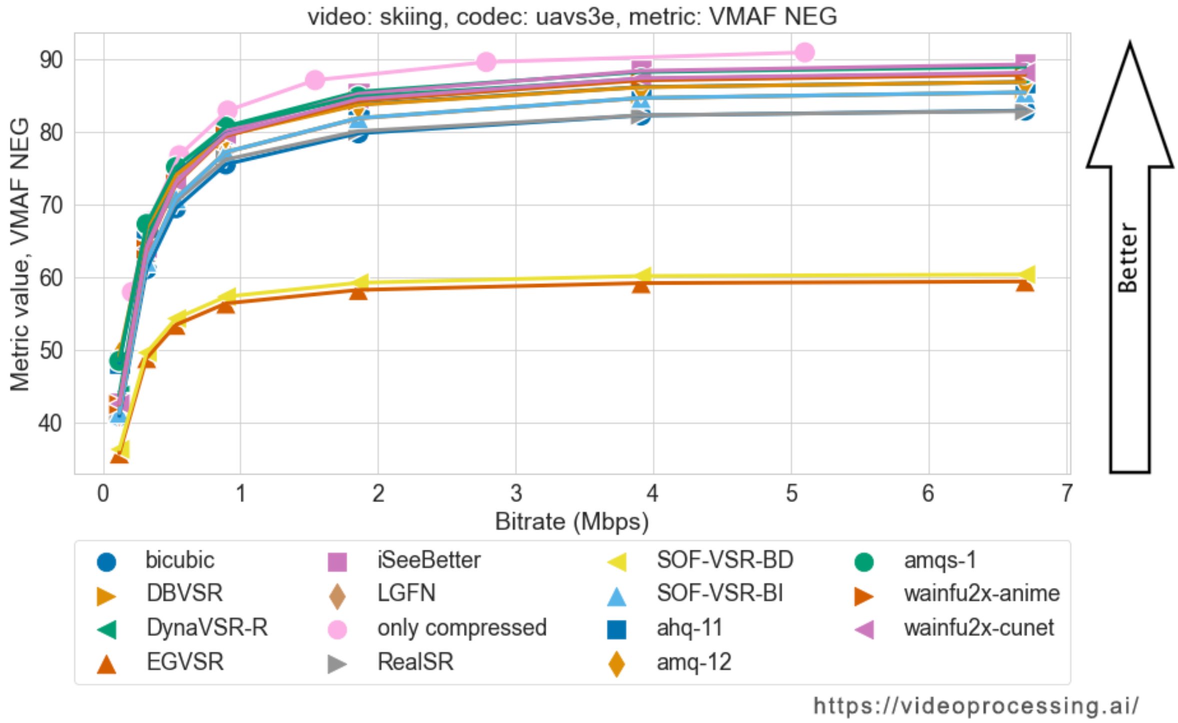


Figure 56a: Bitrate/Quality — *skiing* sequence, uavs3e codec, VMAF NEG metric

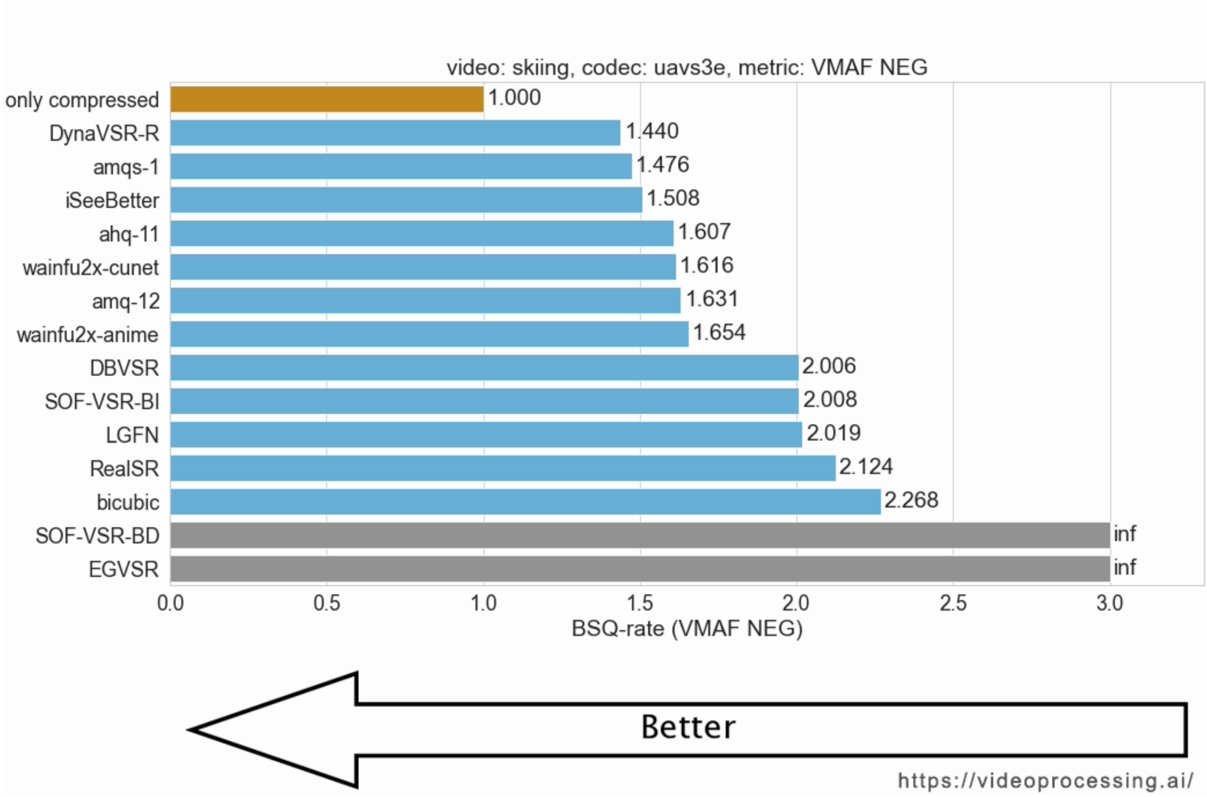


Figure 56b: BSQ-rate relative to “only compressed” —
skiing sequence, uavs3e codec, VMAF NEG metric

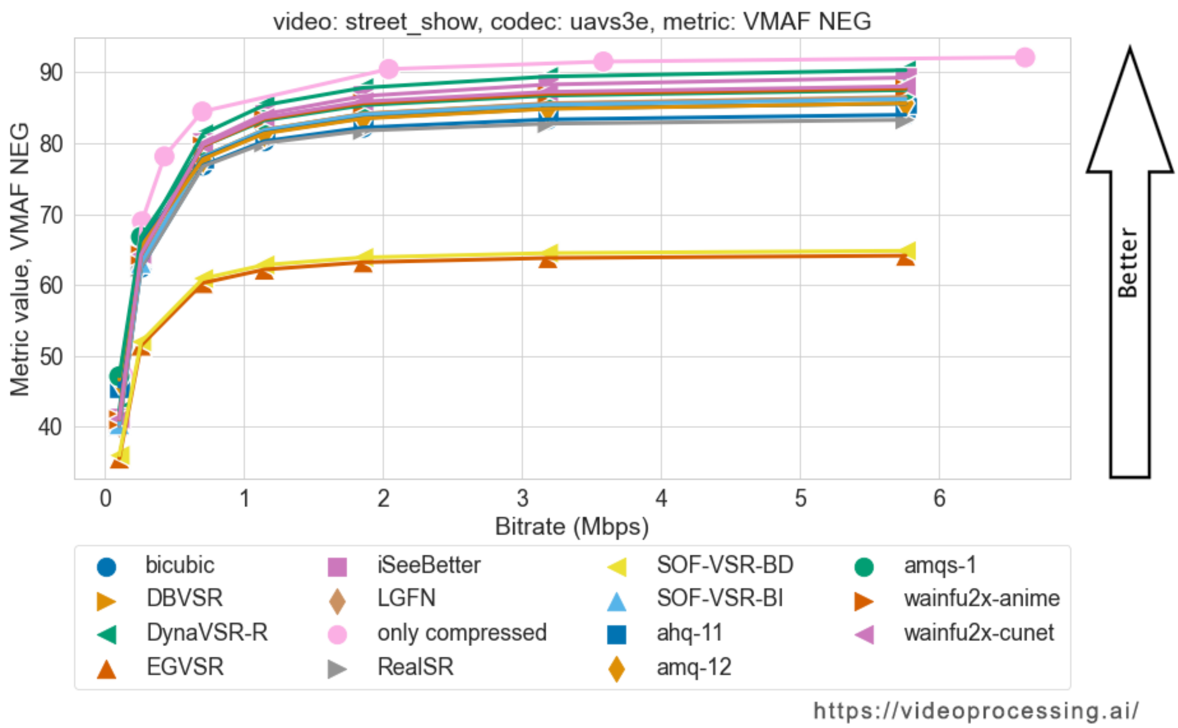


Figure 57a: Bitrate/Quality — *street_show* sequence, uavs3e codec, VMAF NEG metric

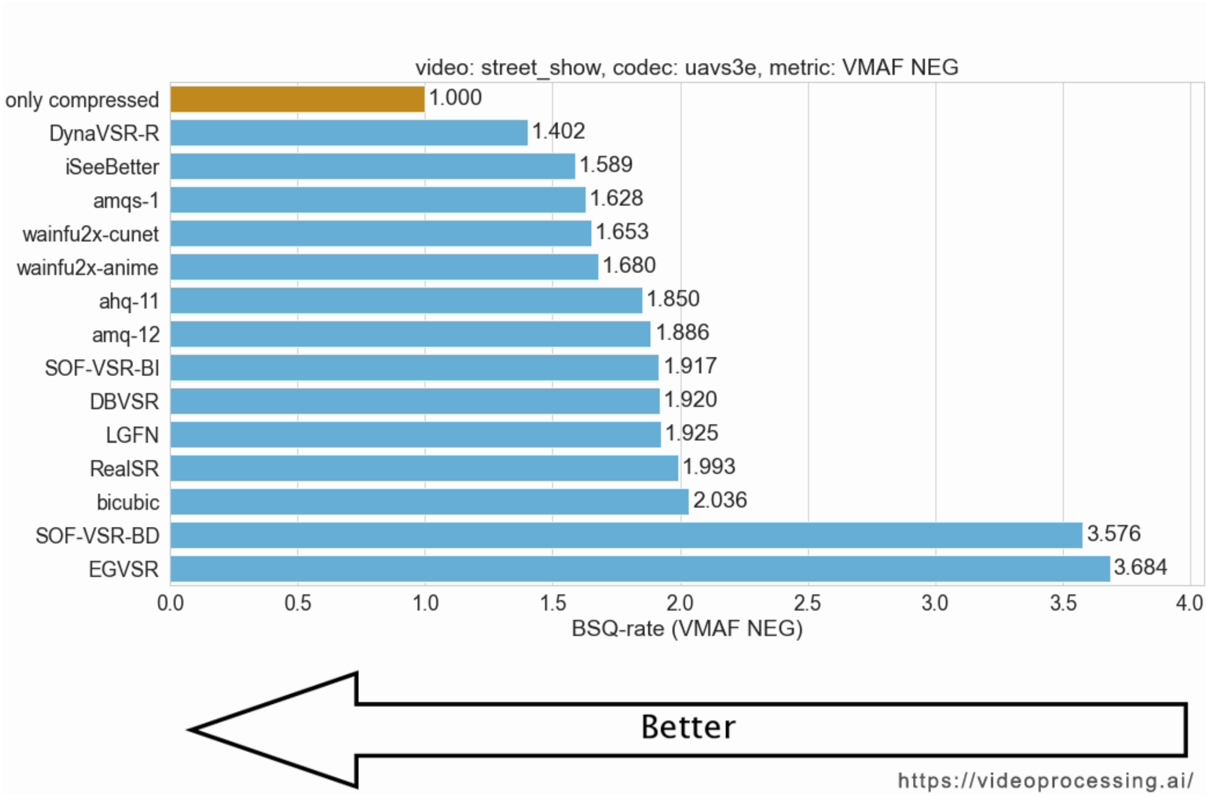


Figure 57b: BSQ-rate relative to “only compressed” —
street_show sequence, uavs3e codec, VMAF NEG metric

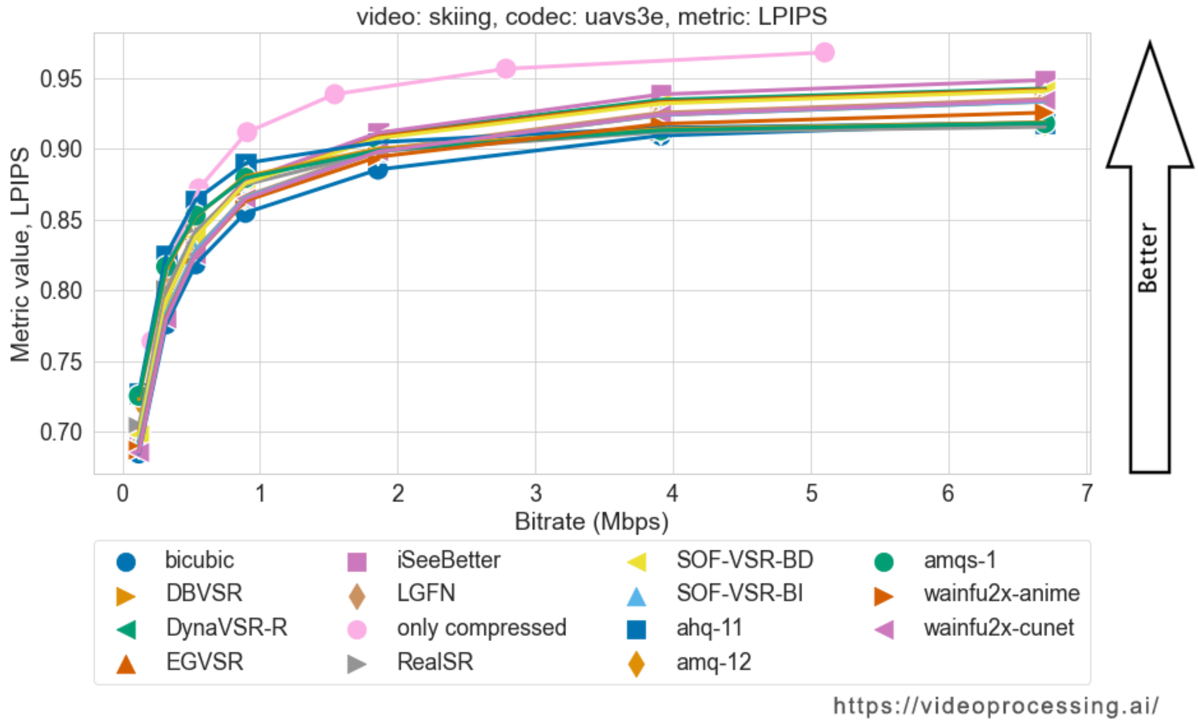


Figure 58a: Bitrate/Quality — *skiing* sequence, uavs3e codec, LPIPS metric

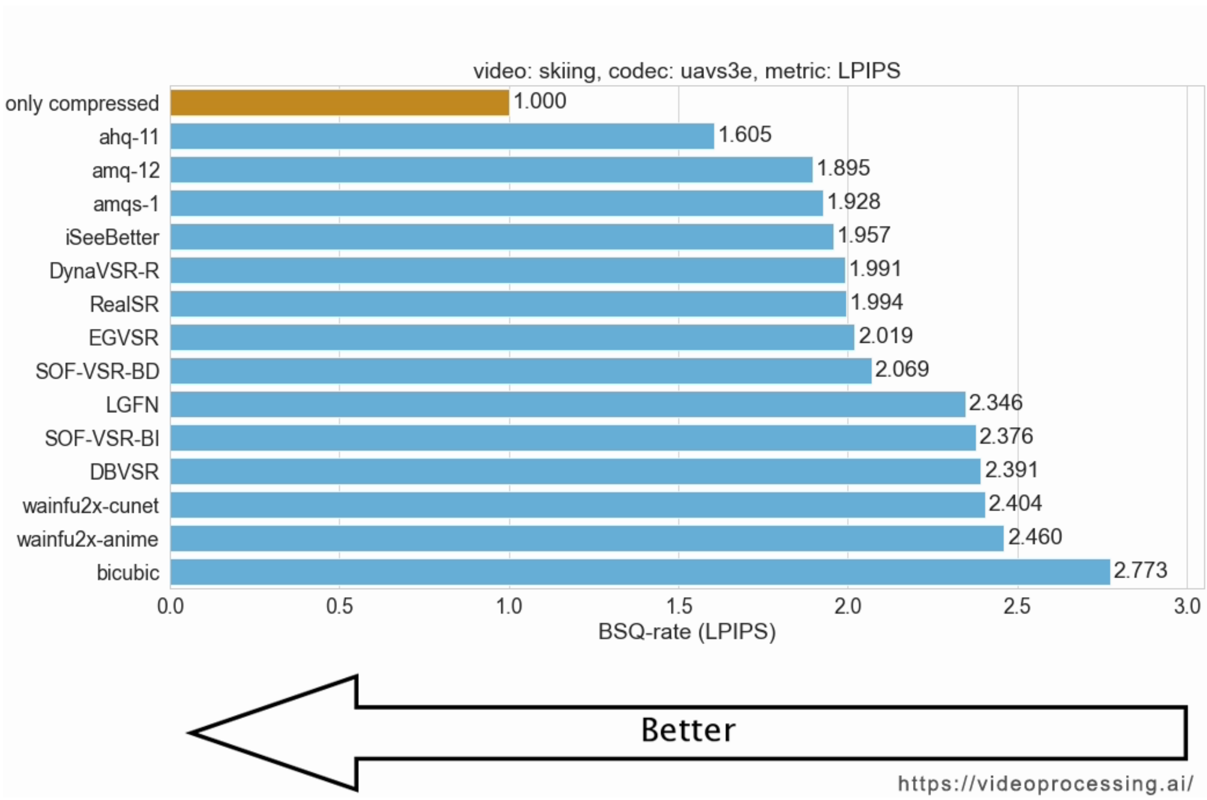


Figure 58b: BSQ-rate relative to “only compressed” —
skiing sequence, uavs3e codec, LPIPS metric

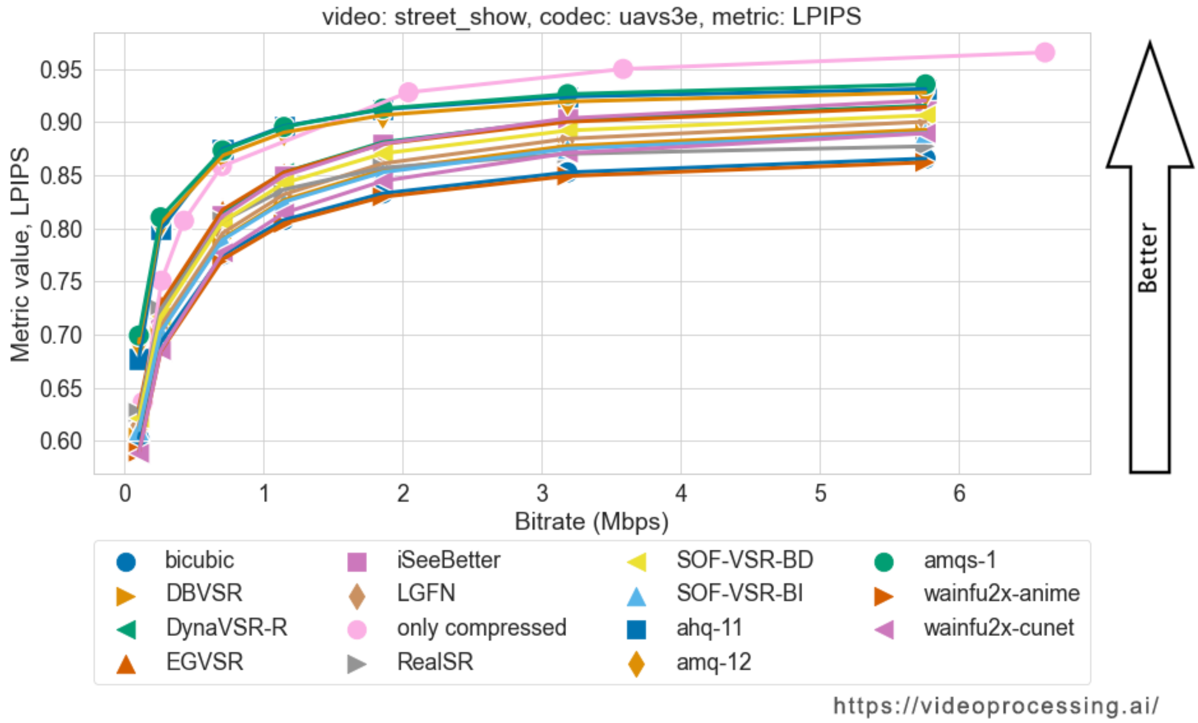


Figure 59a: Bitrate/Quality — *street_show* sequence, uavs3e codec, LPIPS metric

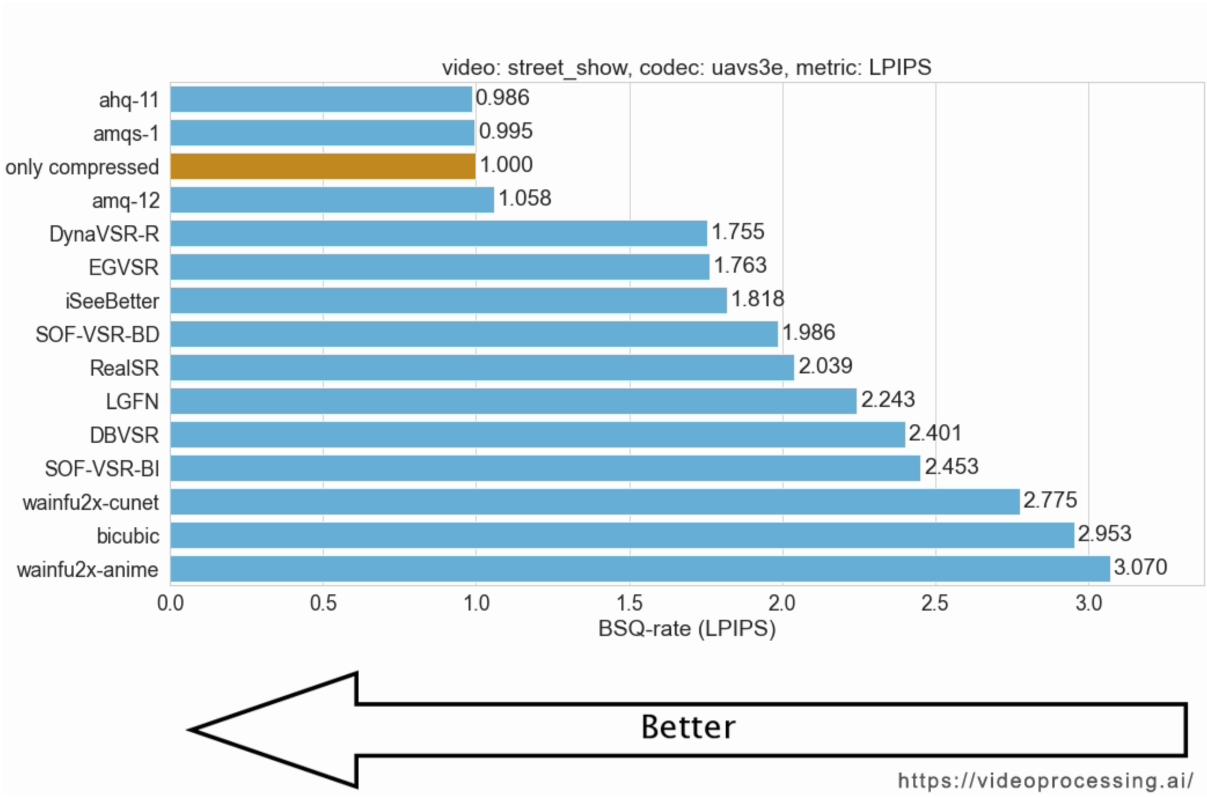


Figure 59b: BSQ-rate relative to “only compressed” —
street_show sequence, uavs3e codec, LPIPS metric

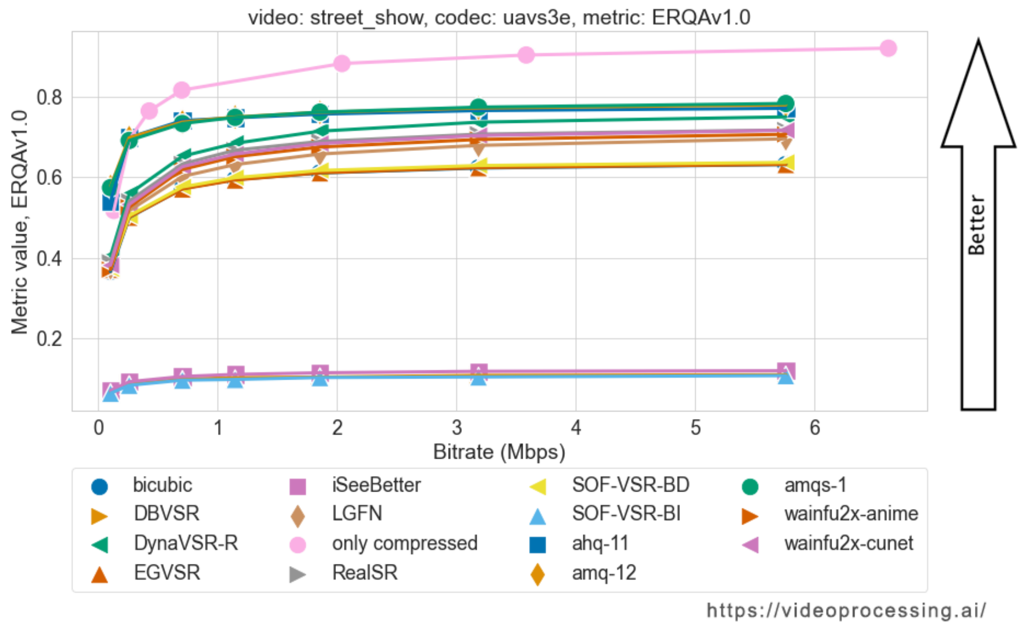


Figure 60a: Bitrate/Quality — *street_show* sequence, uavs3e codec, ERQAv1.0 metric

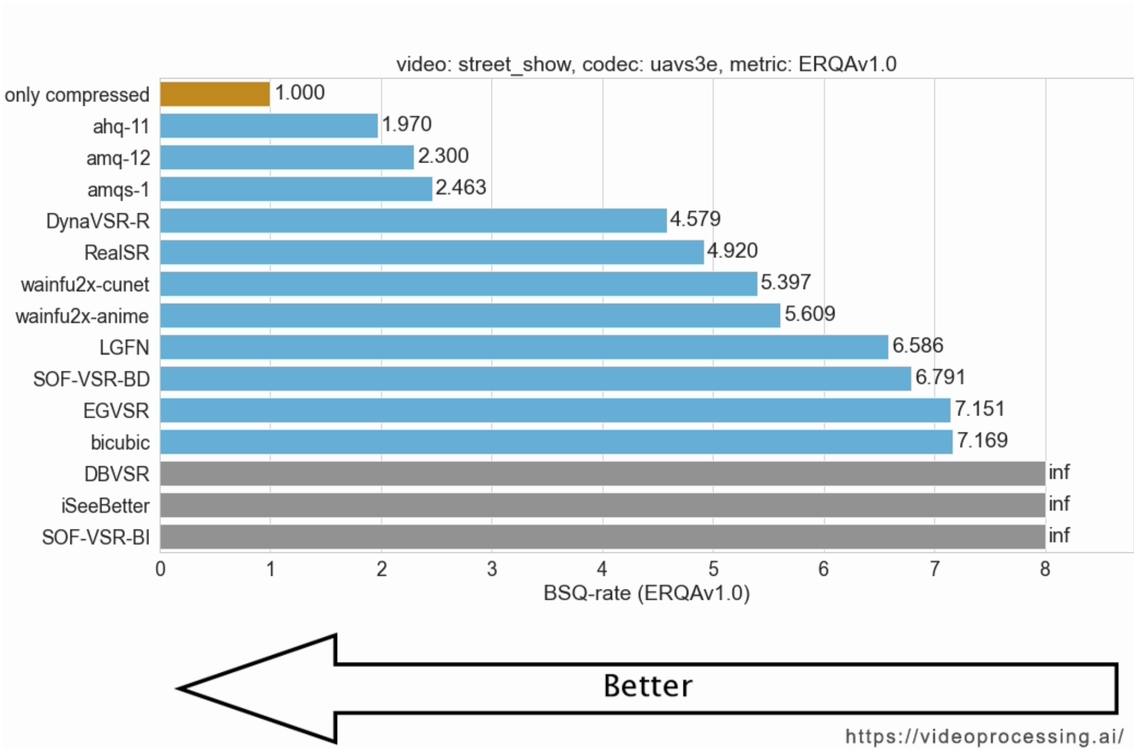


Figure 60b: BSQ-rate relative to “only compressed” — *street_show* sequence, uavs3e codec, ERQAv1.0 metric

In Figure 61 you can see the average BSQ-rate over each metric for the uavs3e codec. “Only compressed” made by the x264 codec was used as a reference.

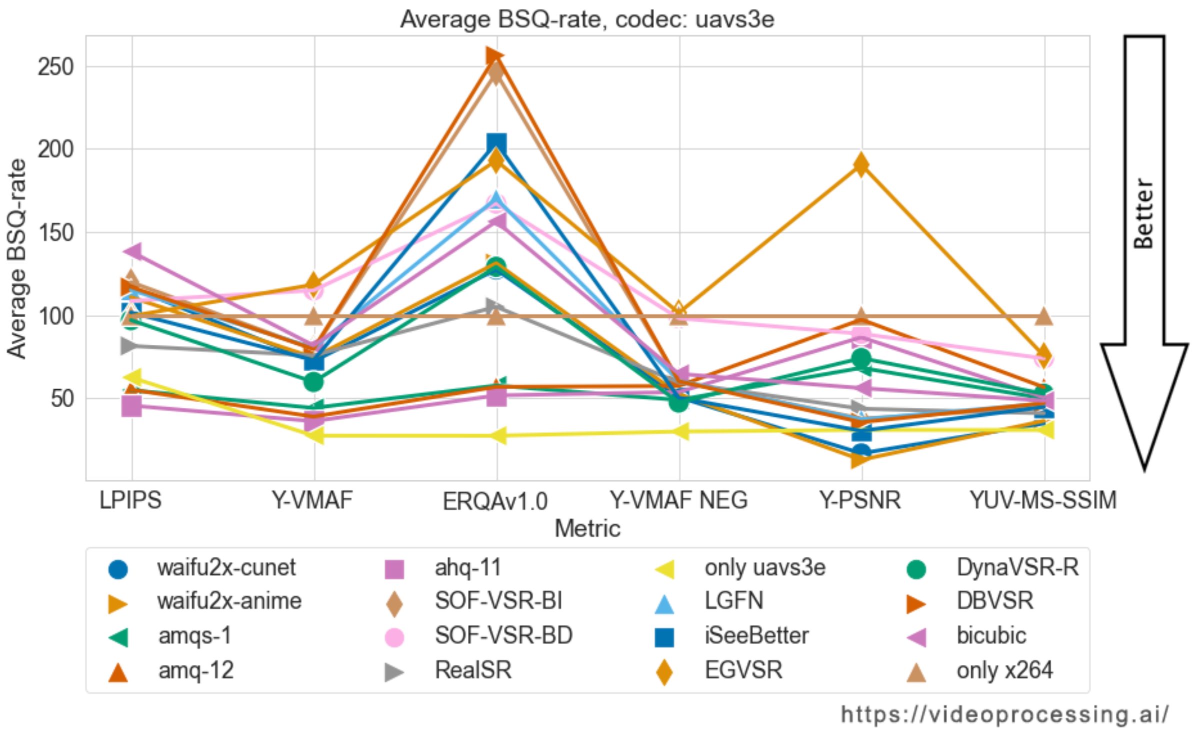


Figure 61: Average BSQ-rate relative to “only x264”.
SR input was compressed with the uavs3e codec

As we can see, some SR models can outperform “only compressed” when working with x264 or x265 codec. However, modern codecs like VVenC outperform SR+codec pairs judging by metrics that correlate well with subjective score (e.g. VMAF, ERQAv1.0).

3. VISUAL/SUBJECTIVE COMPARISON

3.1. Methodology

For subjective comparison, we have chosen 3 different bitrates (1000, 2000, 4000 kbps) from the results of SR models working with the x264 codec. We trim video sequences of 24 frames and convert them to videos with 8 FPS by FFmpeg. Then we took 2 crops with resolution 320×270 from each video and conducted a side-by-side subjective comparison for all these pieces by the subjectify.us platform. Crops were chosen so that the PSNR value on them was close to the average PSNR value over the entire image.

Each one of 1934 accessors had seen 25 paired videos and had to choose which one of them is clearer (option “indistinguishable” is also available). There were 3 verification questions to protect against random answers and bots. In total, we received 57943 answers from valid accessors and used these answers to predict the ranking using the Bradley-Terry model.

To calculate subjective BSQ-rate we extrapolate subjective results using the most similar objective metric. To do this we take the subjective assessments for each test case and find the objective metric that has the highest Pearson correlation with it on the same bitrates. Then we extrapolate a subjective metric using the chosen objective metric as a reference (see Figures 62a and 62b).

Right now we have no evidence that this method of extrapolation is accurate, however, we plan to research this subject in the nearest future.

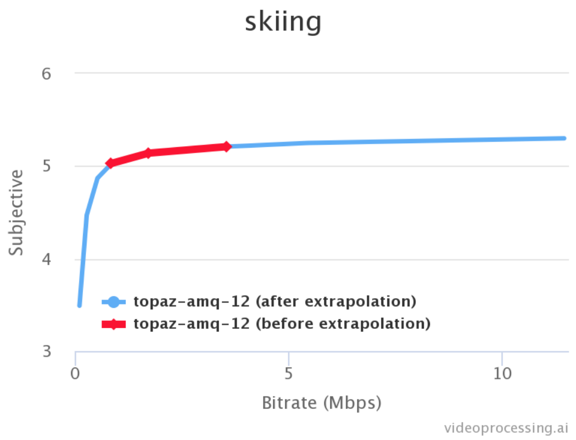


Figure 62a: Subjective metric extrapolation

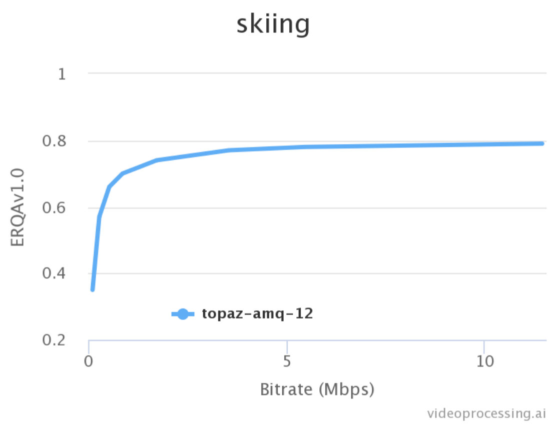


Figure 62b: The most similar objective metric

3.2. Results

The suffixes "_1" and "_2" in video sequences' names mean the number of a cropped segment taken from the original sequence. For example, *animation_clip_1* and *animation_clip_2* are the segments from the *animation_clip* sequence.

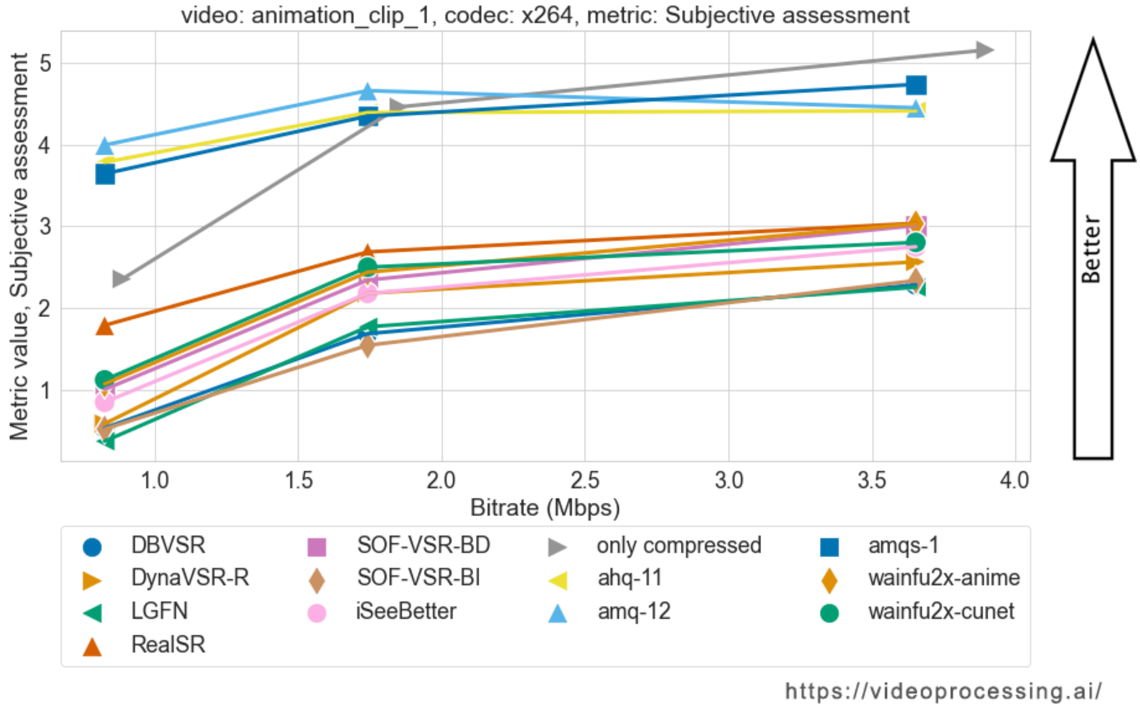


Figure 63a: Bitrate/Quality — *animation_clip_1* sequence, x264 codec, Subjective assessment

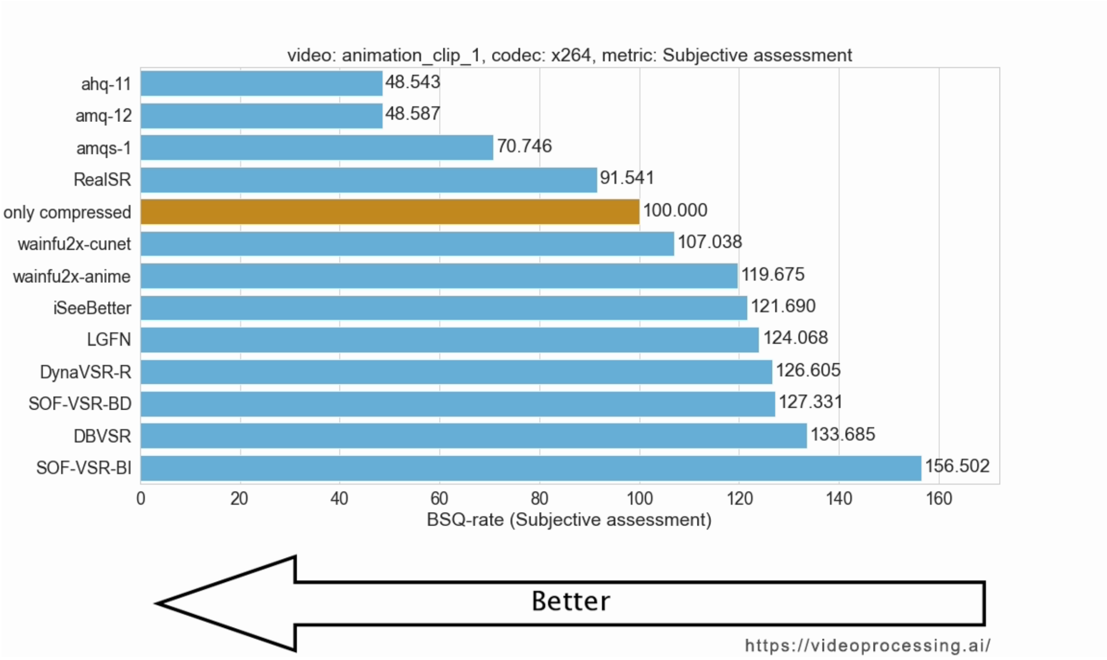


Figure 63b: BSQ-rate relative to "only compressed" —
animation_clip_1 sequence, x264 codec, Subjective assessment

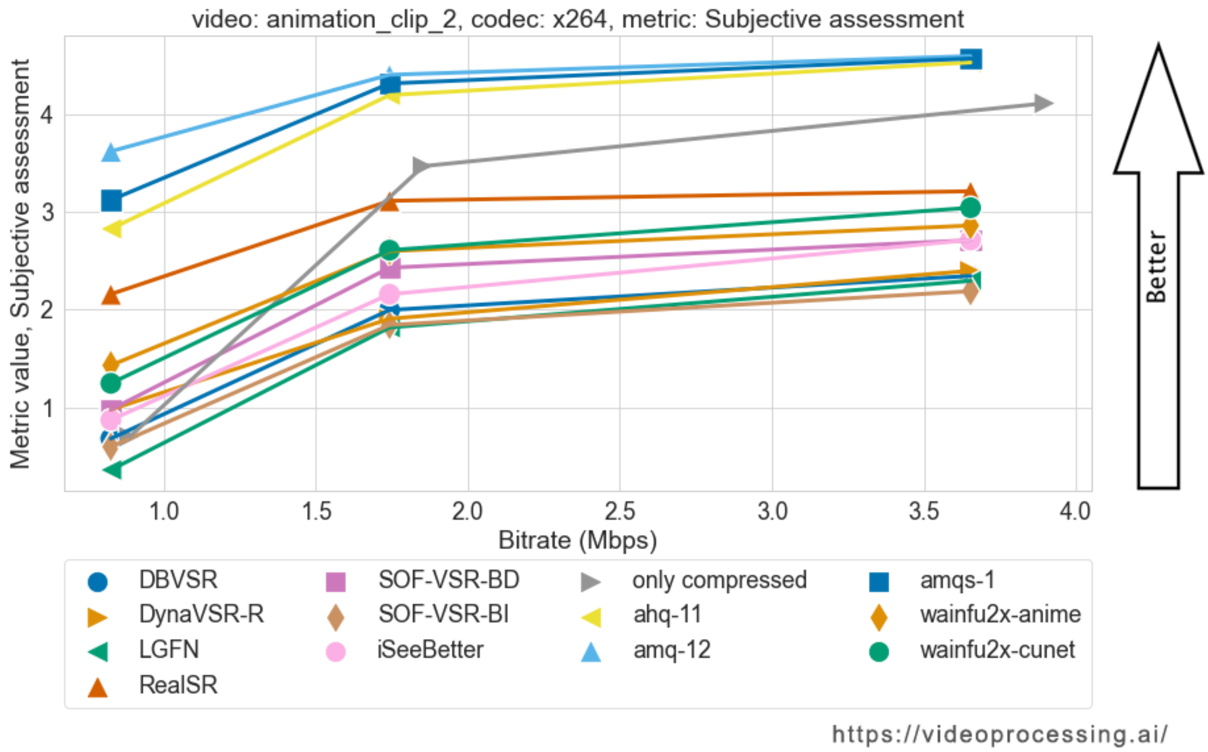


Figure 64a: Bitrate/Quality — *animation_clip_2* sequence, x264 codec, Subjective assessment

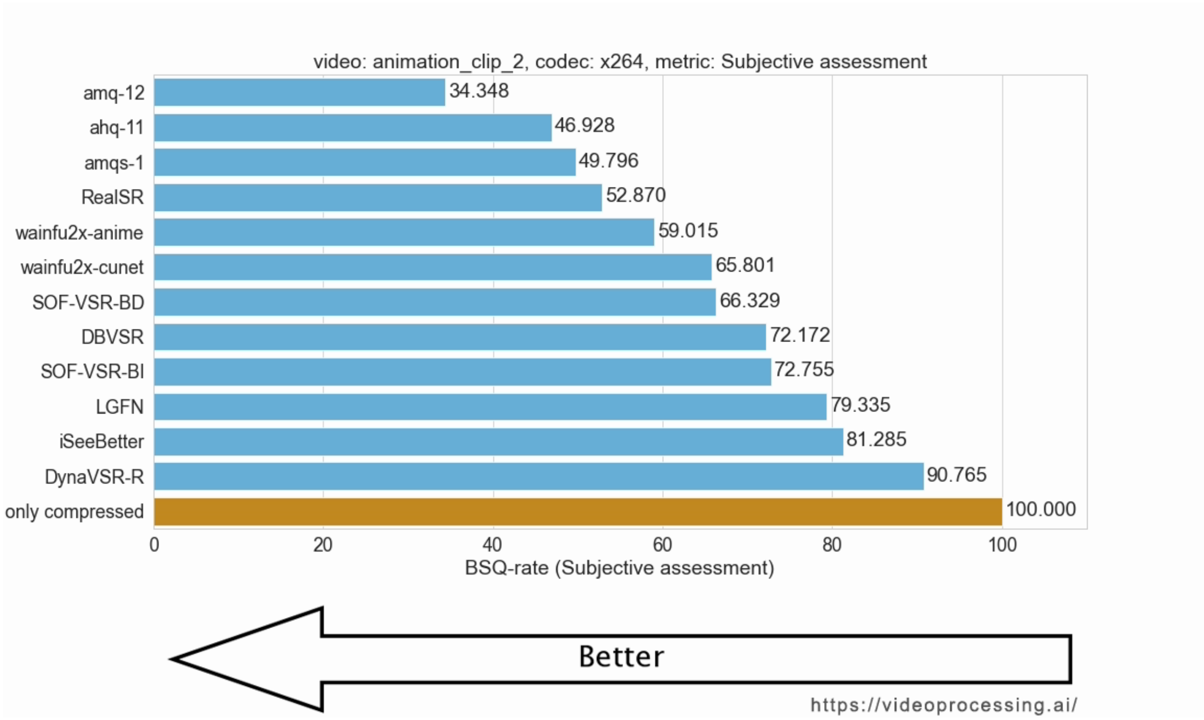


Figure 64b: BSQ-rate relative to “only compressed” — *animation_clip_2* sequence, x264 codec, Subjective assessment

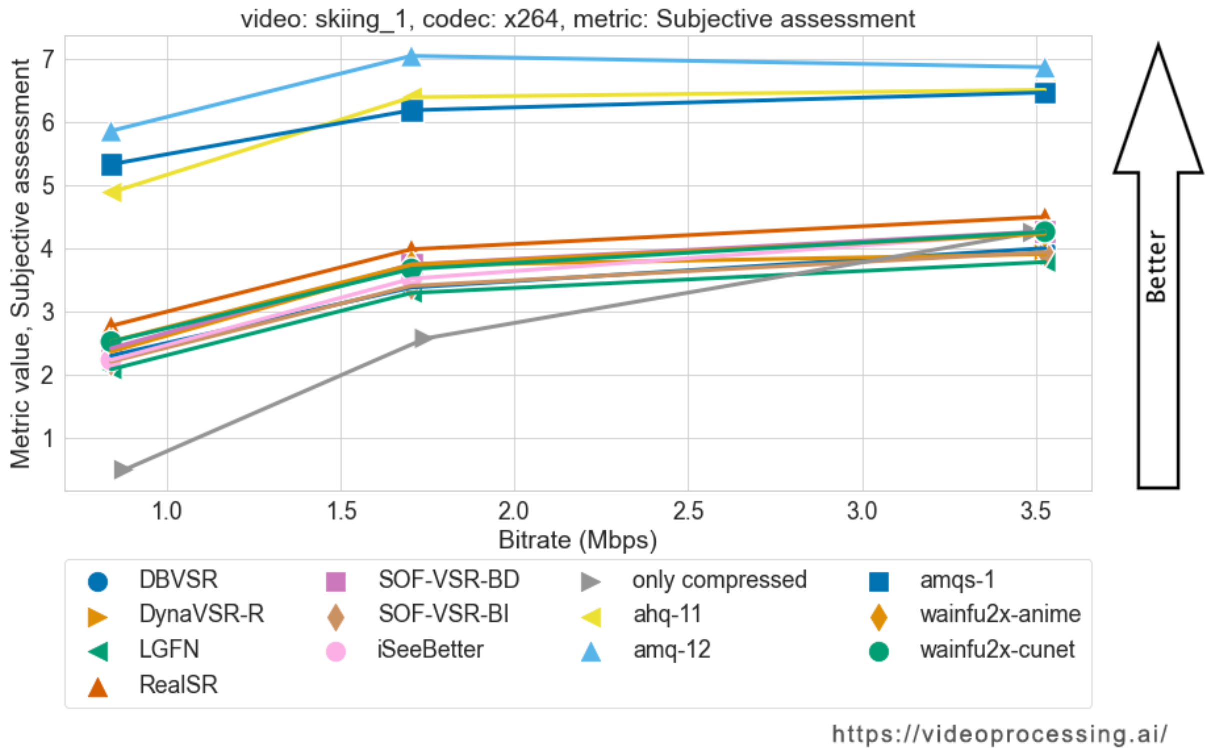


Figure 65a: Bitrate/Quality — *skiing_1* sequence, x264 codec, Subjective assessment

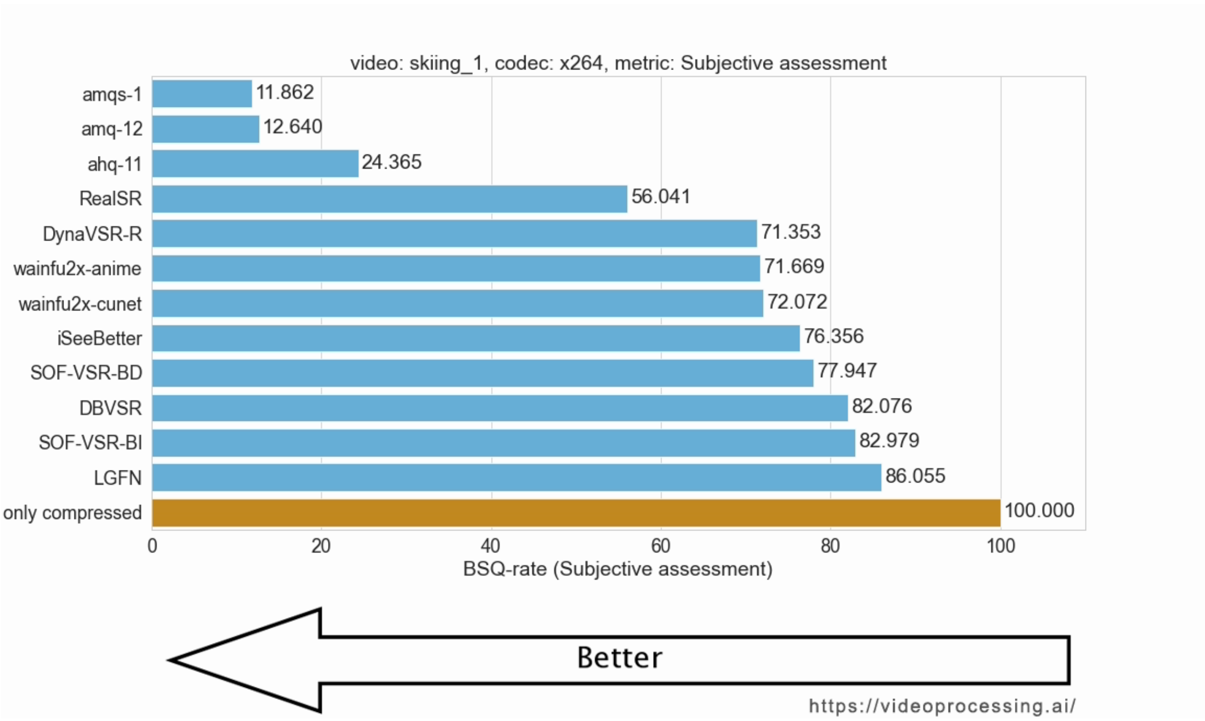


Figure 65b: BSQ-rate relative to “only compressed” —
skiing_1 sequence, x264 codec, Subjective assessment

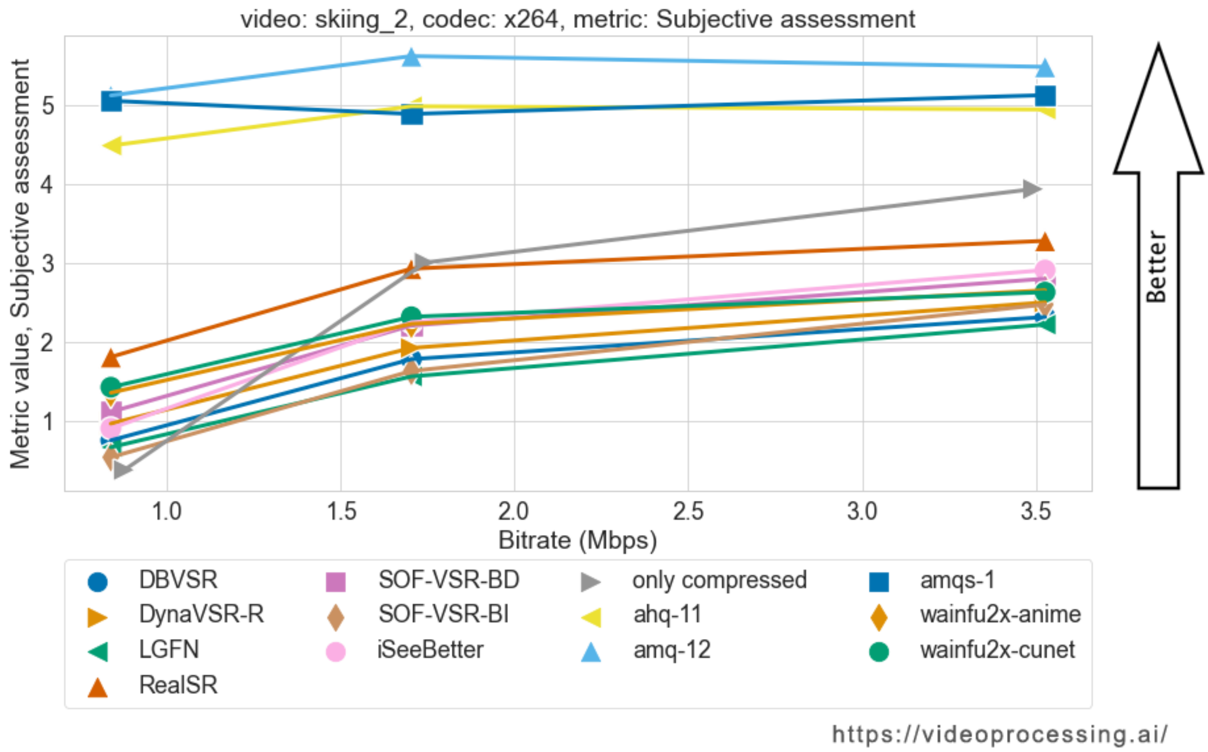


Figure 66a: Bitrate/Quality — *skiing_2* sequence, x264 codec, Subjective assessment

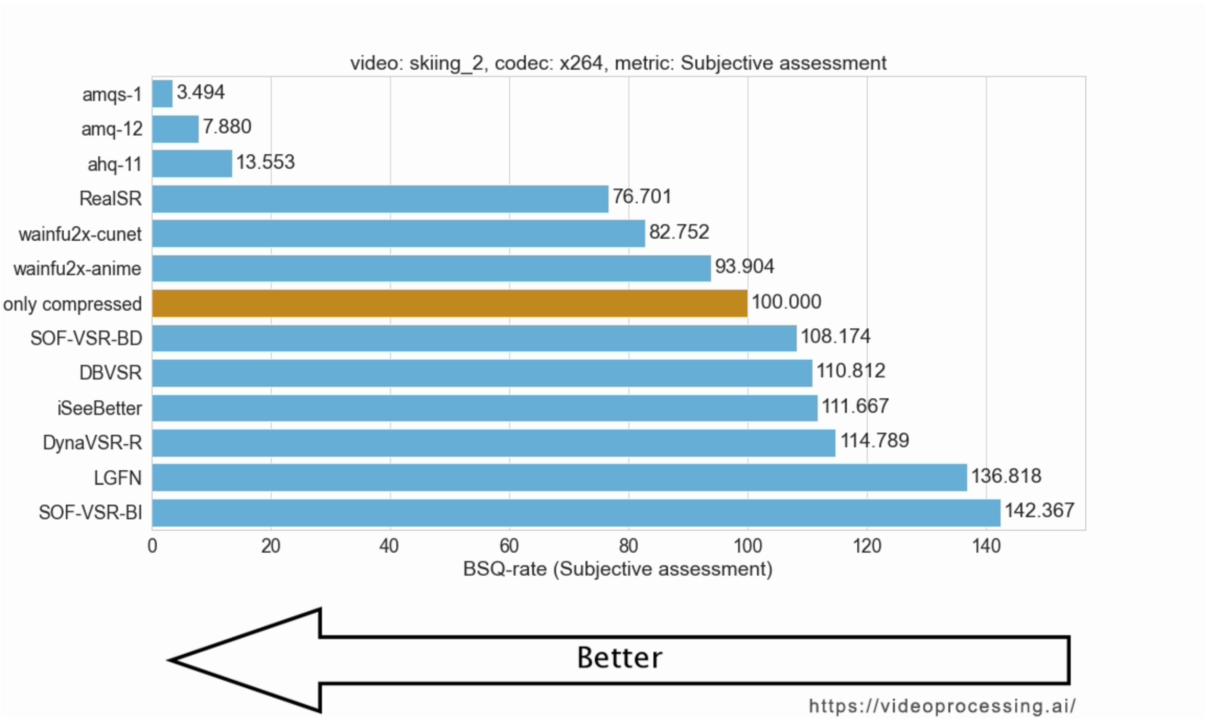


Figure 66b: BSQ-rate relative to "only compressed" —
skiing_2 sequence, x264 codec, Subjective assessment

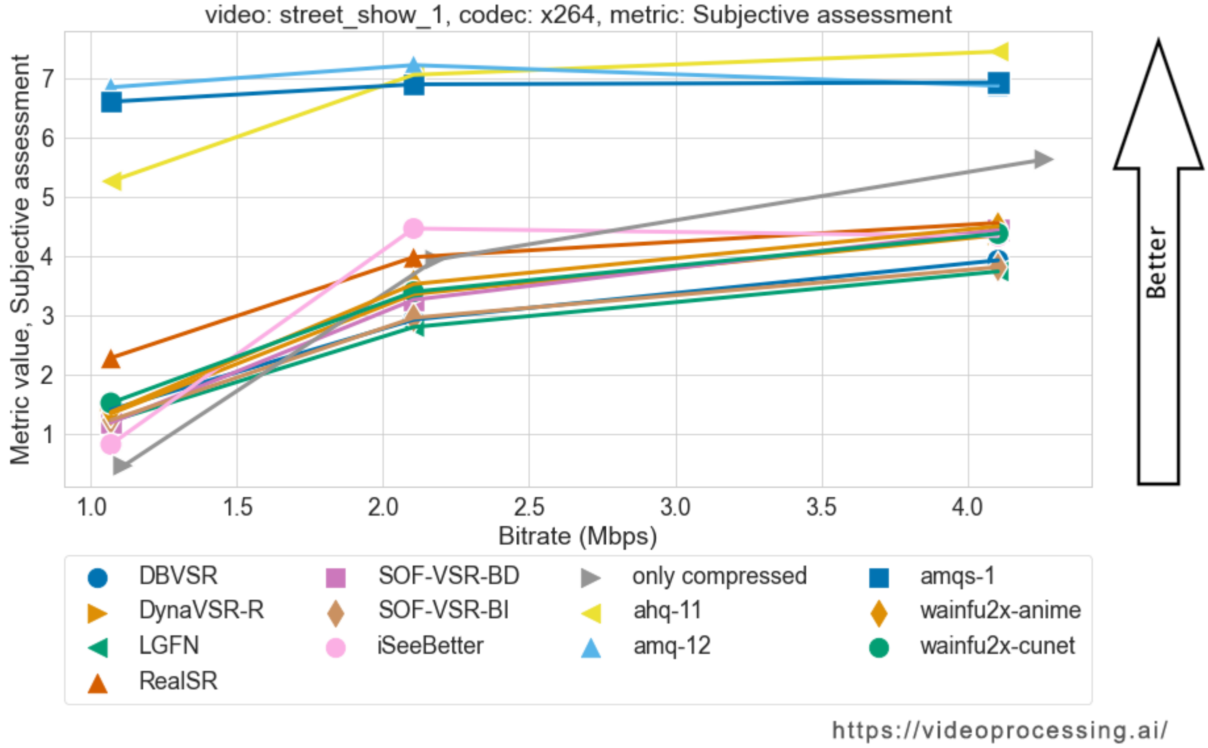


Figure 67a: Bitrate/Quality — *street_show_1* sequence, x264 codec, Subjective assessment

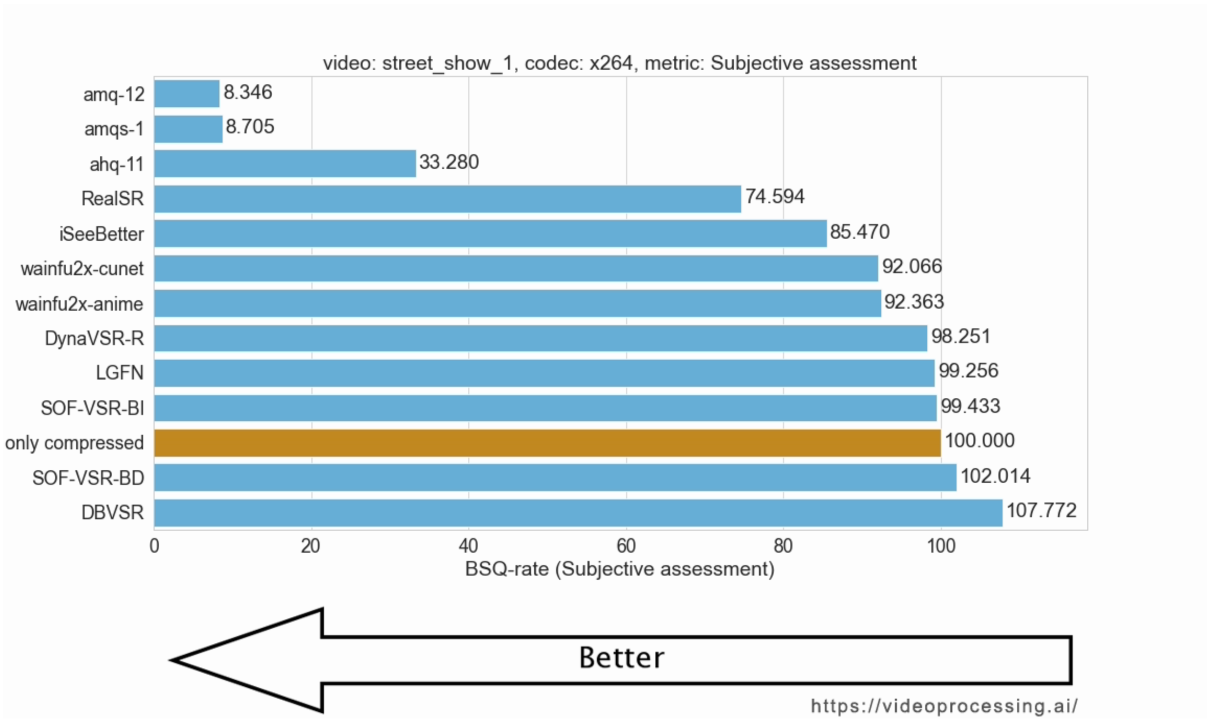


Figure 67b: BSQ-rate relative to “only compressed” — *street_show_1* sequence, x264 codec, Subjective assessment

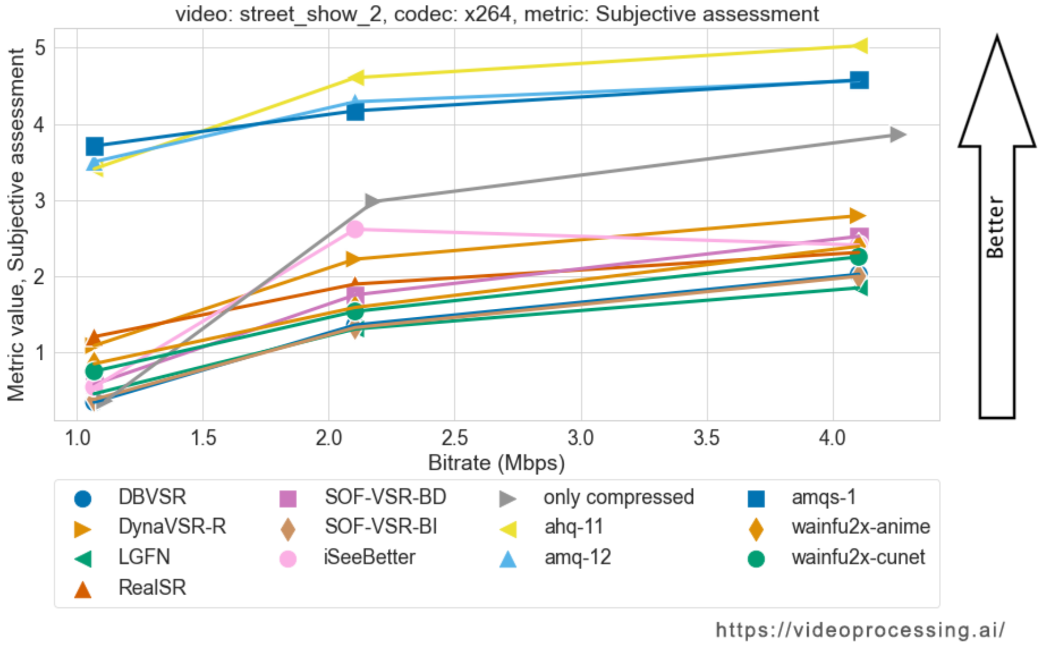


Figure 68a: Bitrate/Quality — *street_show_2* sequence, x264 codec, Subjective assessment

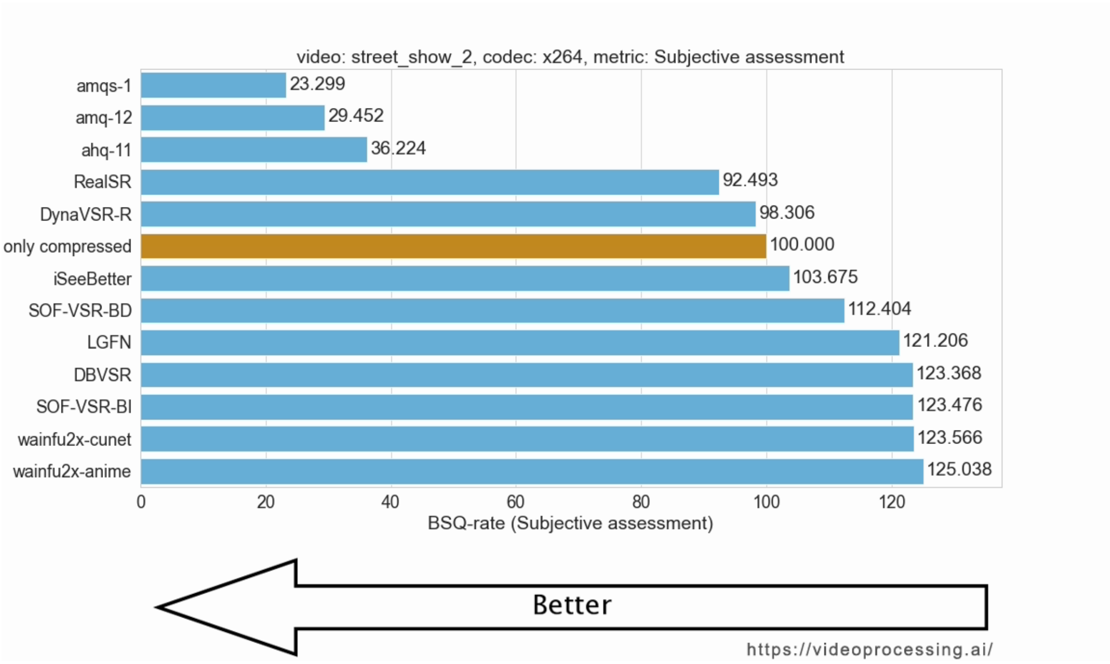


Figure 68b: BSQ-rate relative to “only compressed” —
street_show_2 sequence, x264 codec, Subjective assessment

The subjective comparison showed that commercial filters (ahq-11, amq-12, amqs-1) outperform open-source SR models as well as “only compressed” made by the x264 codec.

4. LEADERBOARDS

The next few tables show a comparison of all pairs of SR algorithms and codecs.
There are also "only compressed" methods (e.g. "only x264"), where we compressed videos without downscaling and upscaling by SR.

For all SR+codecs pairs, BSQ-rate was calculated relative to the “only compressed” made by the x264 codec.

All tables include only the first 25 SR+codec pairs. More information can be found on the benchmark’s website.

4.1. Y-PSNR Leaderboard

Rank	SR + codec	BSQ-rate rel. x264
1	waifu2x-anime + VVenC	10.184
2	waifu2x-anime + uavs3e	12.880
3	waifu2x-cunet + VVenC	13.792
4	waifu2x-cunet + aomenc	14.661
5	waifu2x-anime + aomenc	16.204
6	waifu2x-cunet + uavs3e	16.783
7	waifu2x-anime + x265	20.384
8	waifu2x-cunet + x265	24.885
9	iSeeBetter + uavs3e	30.328
10	only uavs3e	30.856
11	only aomenc	31.638
12	iSeeBetter + aomenc	32.705
13	waifu2x-anime + x264	34.335
14	DBVSR + uavs3e	35.525
15	LGFN + uavs3e	37.199
16	SOF-VSR-BI + uavs3e	37.629
17	waifu2x-cunet + x264	39.203
18	DBVSR + aomenc	40.188
19	RealSR + uavs3e	43.610
20	DBVSR + VVenC	47.448
21	iSeeBetter + x265	47.905
22	LGFN + aomenc	48.013
23	SOF-VSR-BI + aomenc	48.048
24	SOF-VSR-BI + VVenC	49.137
25	RealSR + aomenc	49.670

Table 4: Leaderboard, Y-PSNR metric

4.2. YUV-MS-SSIM Leaderboard

Rank	SR + codec	BSQ-rate rel. x264
1	waifu2x-anime + VVenC	25.248
2	waifu2x-cunet + VVenC	25.346
3	only aomenc	28.474
4	iSeeBetter + VVenC	30.110
5	SOF-VSR-BI + VVenC	30.690
6	only uavs3e	30.820
7	LGFN + VVenC	31.952
8	RealSR + VVenC	33.013
9	waifu2x-cunet + aomenc	33.506
10	DynaVSR-R + VVenC	33.649
11	DBVSR + VVenC	34.499
12	waifu2x-cunet + uavs3e	34.793
13	waifu2x-anime + aomenc	35.114
14	waifu2x-anime + uavs3e	36.295
15	amqs-1 + VVenC	40.103
16	RealSR + uavs3e	40.914
17	SOF-VSR-BI + aomenc	43.697
18	SOF-VSR-BI + uavs3e	44.424
19	iSeeBetter + uavs3e	44.735
20	iSeeBetter + aomenc	44.797
21	RealSR + aomenc	46.462
22	amq-12 + VVenC	46.601
23	LGFN + aomenc	46.747
24	LGFN + uavs3e	46.764
25	DBVSR + uavs3e	47.015

Table 5: Leaderboard, YUV-MS-SSIM metric

4.3. Y-VMAF Leaderboard

Rank	SR + codec	BSQ-rate rel. x264
1	only aomenc	26.454
2	only uavs3e	27.410
3	ahq-11 + uavs3e	36.128
4	ahq-11 + VVenC	36.260
5	ahq-11 + aomenc	37.225
6	amq-12 + VVenC	37.895
7	amq-12 + uavs3e	38.880
8	amq-12 + aomenc	41.220
9	amqs-1 + uavs3e	44.088
10	amqs-1 + VVenC	44.988
11	amqs-1 + aomenc	48.057
12	DynaVSR-R + VVenC	51.299
13	only VVenC	52.612
14	ahq-11 + x265	55.192
15	amq-12 + x265	55.458
16	only x265	59.551
17	DynaVSR-R + uavs3e	59.739
18	amqs-1 + x265	60.579
19	waifu2x-cunet + VVenC	62.987
20	iSeeBetter + VVenC	63.063
21	DynaVSR-R + aomenc	63.183
22	waifu2x-anime + VVenC	63.581
23	RealSR + VVenC	63.719
24	DBVSR + VVenC	66.249
25	SOF-VSR-BI + VVenC	67.477

Table 6: Leaderboard, Y-VMAF metric

4.4. Y-VMAF NEG Leaderboard

Rank	SR + codec	BSQ-rate rel. x264
1	only uavs3e	29.909
2	only aomenc	30.172
3	DynaVSR-R + VVenC	39.539
4	only VVenC	43.422
5	iSeeBetter + VVenC	44.865
6	waifu2x-cunet + VVenC	45.658
7	waifu2x-anime + VVenC	45.862
8	DynaVSR-R + uavs3e	47.380
9	amqs-1 + VVenC	47.960
10	amqs-1 + uavs3e	48.921
11	iSeeBetter + uavs3e	50.245
12	waifu2x-cunet + uavs3e	50.570
13	RealSR + VVenC	51.195
14	SOF-VSR-BI + VVenC	52.234
15	waifu2x-anime + uavs3e	52.427
16	DynaVSR-R + aomenc	52.537
17	amq-12 + VVenC	52.710
18	DBVSR + VVenC	52.985
19	LGFN + VVenC	53.063
20	ahq-11 + uavs3e	53.679
21	ahq-11 + VVenC	54.215
22	amqs-1 + aomenc	55.718
23	amq-12 + uavs3e	57.337
24	iSeeBetter + aomenc	57.552
25	waifu2x-cunet + aomenc	58.282

Table 7: Leaderboard, Y-VMAF NEG metric

4.5. LPIPS Leaderboard

Rank	SR + codec	BSQ-rate rel. x264
1	ahq-11 + uavs3e	45.543
2	ahq-11 + VVenC	46.160
3	ahq-11 + aomenc	48.276
4	ahq-11 + x265	51.588
5	amq-12 + VVenC	52.454
6	amqs-1 + VVenC	53.027
7	amqs-1 + uavs3e	54.851
8	amq-12 + uavs3e	55.096
9	amqs-1 + x265	58.185
10	amq-12 + x265	58.370
11	only VVenC	58.503
12	amqs-1 + aomenc	58.641
13	amq-12 + aomenc	59.240
14	ahq-11 + x264	62.357
15	only uavs3e	62.480
16	RealSR + VVenC	63.704
17	amq-12 + x264	66.881
18	only x265	67.949
19	amqs-1 + x264	68.373
20	only aomenc	71.213
21	RealSR + x265	81.294
22	RealSR + uavs3e	81.429
23	waifu2x-anime + VVenC	81.715
24	waifu2x-cunet + VVenC	84.355
25	DynaVSR-R + VVenC	85.468

Table 8: Leaderboard, LPIPS metric

4.6. ERQAv1.0 Leaderboard

Rank	SR + codec	BSQ-rate rel. x264
1	only VVenC	24.956
2	only aomenc	26.403
3	only uavs3e	27.373
4	ahq-11 + uavs3e	51.633
5	only x265	56.435
6	amq-12 + uavs3e	56.790
7	amqs-1 + uavs3e	57.595
8	amq-12 + VVenC	59.169
9	amqs-1 + VVenC	59.666
10	ahq-11 + VVenC	60.560
11	ahq-11 + aomenc	62.480
12	ahq-11 + x265	64.839
13	amq-12 + x265	67.095
14	amqs-1 + x265	69.926
15	amqs-1 + aomenc	73.008
16	amq-12 + aomenc	73.416
17	RealSR + VVenC	85.524
18	amq-12 + x264	86.343
19	ahq-11 + x264	89.941
20	amqs-1 + x264	92.334
21	only x264	100.000
22	waifu2x-anime + VVenC	103.208
23	RealSR + uavs3e	105.007
24	waifu2x-cunet + VVenC	107.416
25	DynaVSR-R + VVenC	109.760

Table 9: Leaderboard, ERQAv1.0 metric

5. CORRELATION OF METRICS WITH SUBJECTIVE ASSESSMENT

We calculated Pearson and Spearman correlation of metrics with subjective assessment.

We calculated objective metrics on the crops used for subjective comparison and found a correlation between the subjective and objective results (see Figure 69). We consider the correlation of metric values with subjective assessment separately on all crops and then calculate the mean correlation.

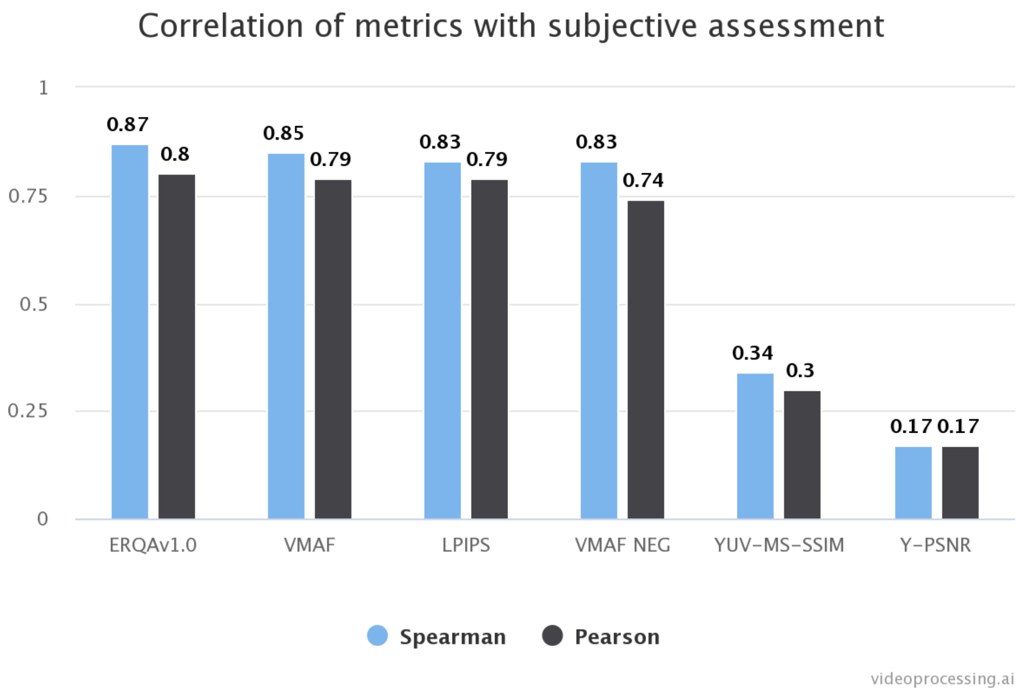


Figure 69: Correlation of metrics on crops

We can see that the most popular metrics for SR — PSNR and SSIM — correlate poorly with subjective assessment. As you can see from Fig. 69, ERQA is superior to other metrics in terms of correlation with the subjective score.

6. SR+CODEC BENCHMARK ROADMAP

Feature	What it achieves	Release date
4 more video sequences	We will extend our dataset to make it more diverse and cover more use cases. We expect it to contain 7×7×5=**245 Full HD videos**.	Q4 2021
Subjective comparison for x265, aomenc, VVenC, and uavs3e codecs	Right now, we have made a subjective comparison for only one codec. We plan to make a subjective comparison for **all codecs** to get a more accurate ranking for more SR+codec pairs. The subjective comparison with that many video pairs will be **very expensive**. If you want to support our benchmark, please contact us: sr-codecs-benchmark@videoprocessing.ai	Q4 2021
More state-of-the-art Super-Resolution methods	New Super-Resolution methods are constantly being developed. We will add new qualitative SR methods to our benchmark as they appear. We also expect developers to submit their methods to us. You can submit your method here.	Q1 2022
A new metric to measure compressed video restoration quality	The subjective comparison showed that the most popular video quality metrics — PSNR and SSIM — are not applicable to the Super-Resolution task. We are researching our metric for compressed video restoration quality that will **correlate well with subjective assessment.**	Q1 2022
“Real-time” and “restoration” categories	Some Super-Resolution models work faster than others, while slow methods can achieve results of much better quality. We plan to divide the leaderboard of our benchmark into 2 categories: • The “**real-time”** category will contain fast SR methods that can be used to enhance videos in real-time • The **“restoration”** category will contain SR methods that can produce high-quality results over any period of time	Q2 2022

A. SEQUENCES

	**Animation Clip** 2D animation advertising clip drawn in bright colors.	Full HD, 100 frames, 30 fps, 104.58 Mbps Tags: *tv ads, animation*
	**Skiing Learning** People are being trained to ski in slow motion.	Full HD, 179 frames, 24 fps, 107.59 Mbps Tags: *double exposure, hand/head-mounted camera*
	**Street Show** Two men sing, dance, and perform some acrobatics on a street.	Full HD, 200 frames, 24 fps, 108.40 Mbps Tags: *aero shooting, flash exposure*

Table 10: Video sequences statistics

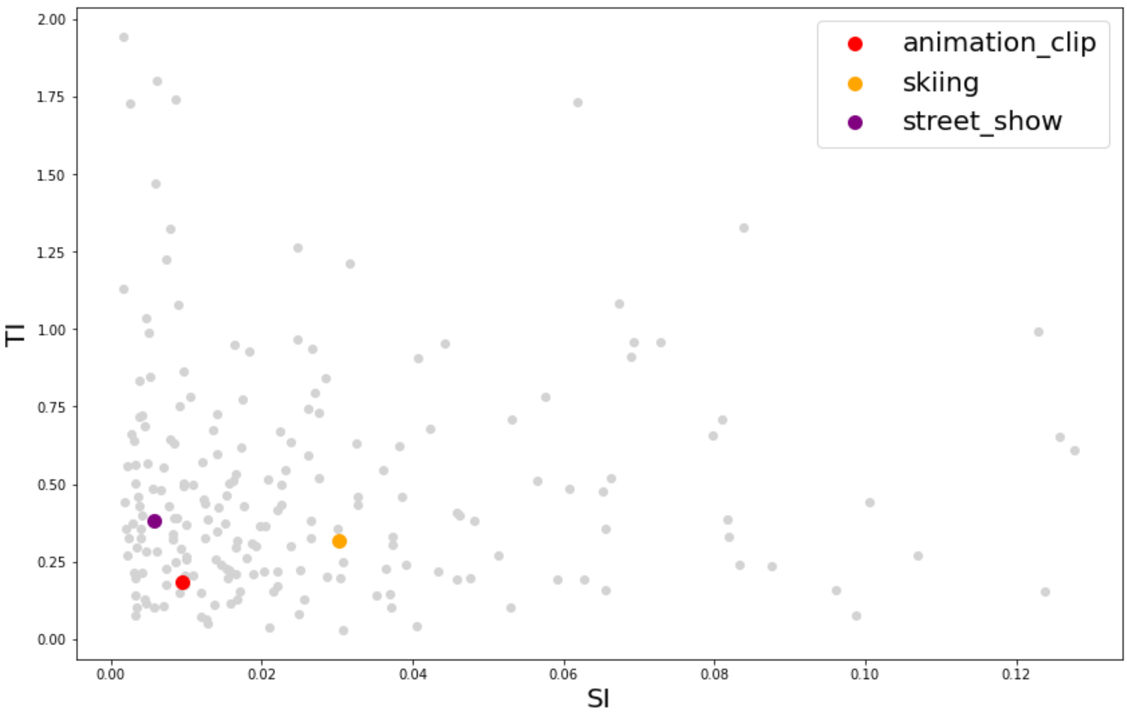


Figure 70: SI/TI of all videos in the [MSU Video Codecs Comparison] (gray dots) and the chosen videos

B. CODECS

B.1. x264

Standard: H.264
Implementation: FFmpeg version 4.2.4
Preset: medium

B.2. x265

Standard: H.265
Implementation: FFmpeg version 4.2.4
Preset: medium

B.3. aomenc

Standard: AV1
Implementation: FFmpeg version 4.2.4
Preset: medium

B.4. VVenC

Standard: H.266
Implementation: https://github.com/fraunhoferhhi/vvenc
Preset: medium
Version: v1.0.0

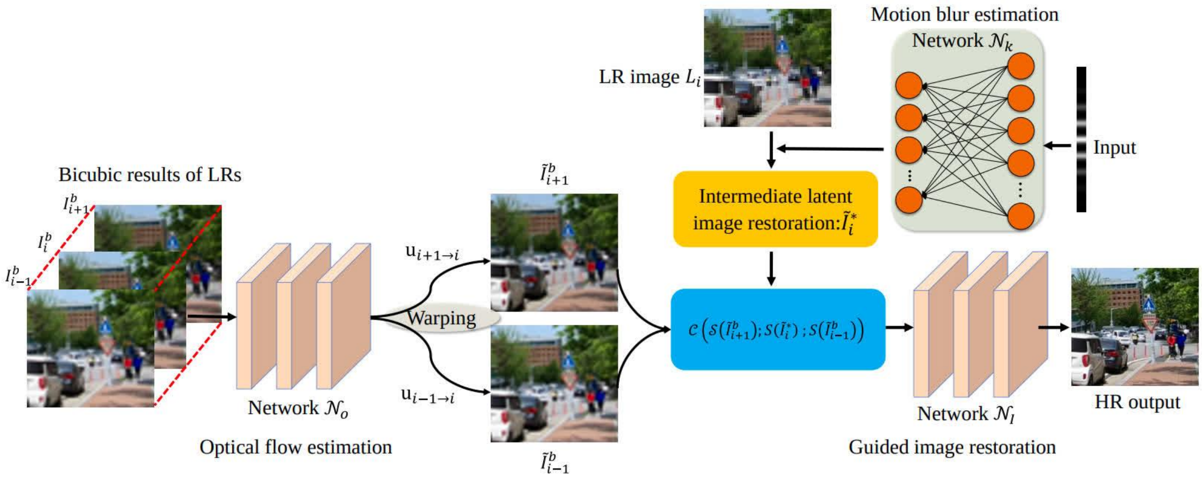
B.5. uavs3e

Standard: AVS3
Implementation: https://github.com/uavs3/uavs3e
Commit: cd29508

C. PARTICIPANTS

C.1. DBVSR

Estimates a motion blur for the particular input. Compensates the motion between frames explicitly.

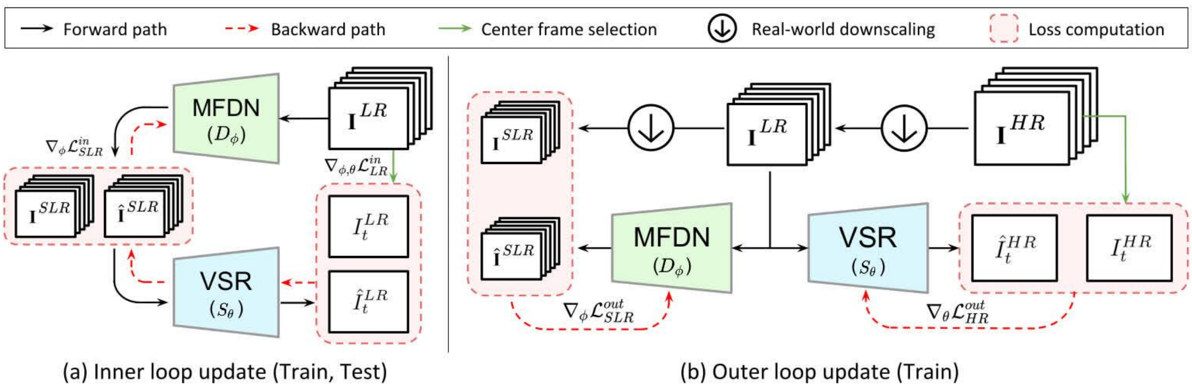


Added to the benchmark by MSU G&M Lab

Links: GitHub, paper

C.2. DynaVSR

Uses meta-learning to estimate a degradation kernel for the particular input.
DynaVSR can be applied to any VSR deep-learning model. For our benchmark, we used pre-trained weights for model EDVR, which uses Deformable convolution to align neighboring frames.

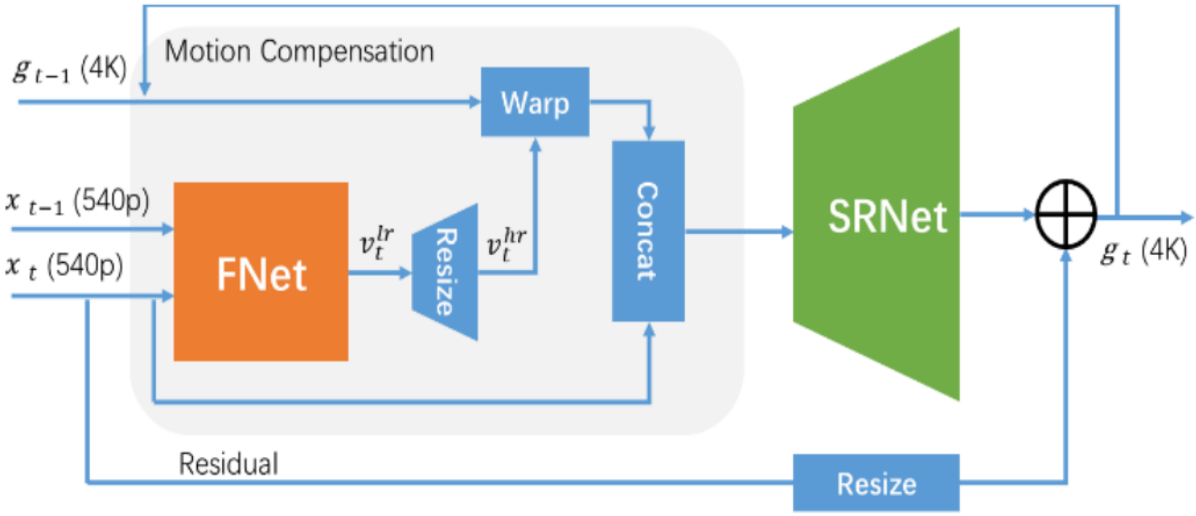


Added to the benchmark by MSU G&M Lab

Links: GitHub, paper

C.3. EGVSR

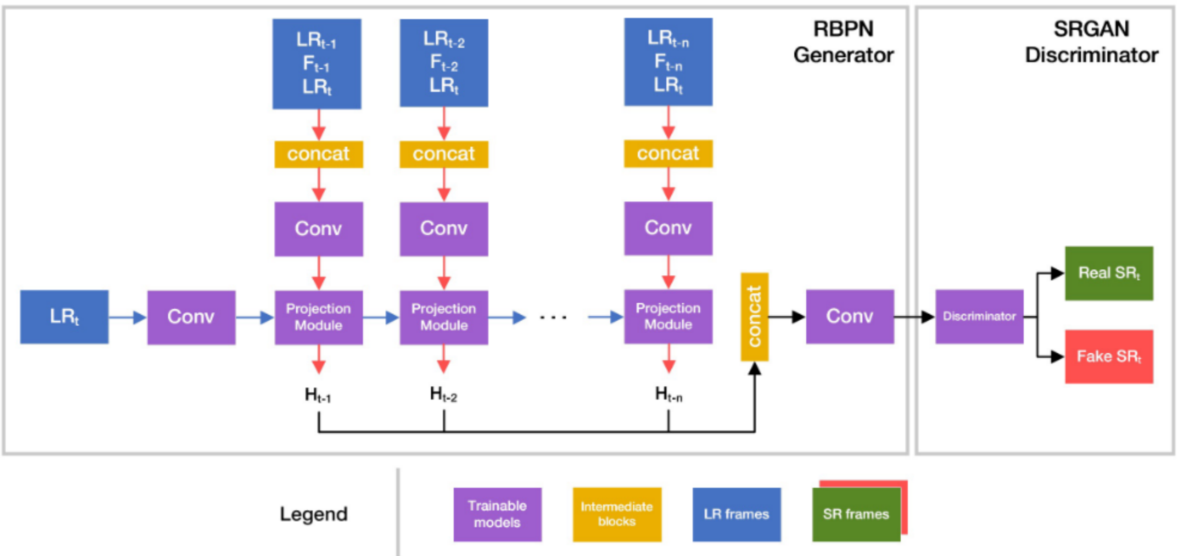
The generator part is divided into FNet module and SRNet module for optical flow estimation and video frame super-resolution, respectively.



Added to the benchmark by MSU G&M Lab
Links: GitHub, paper

C.4. iSeeBetter

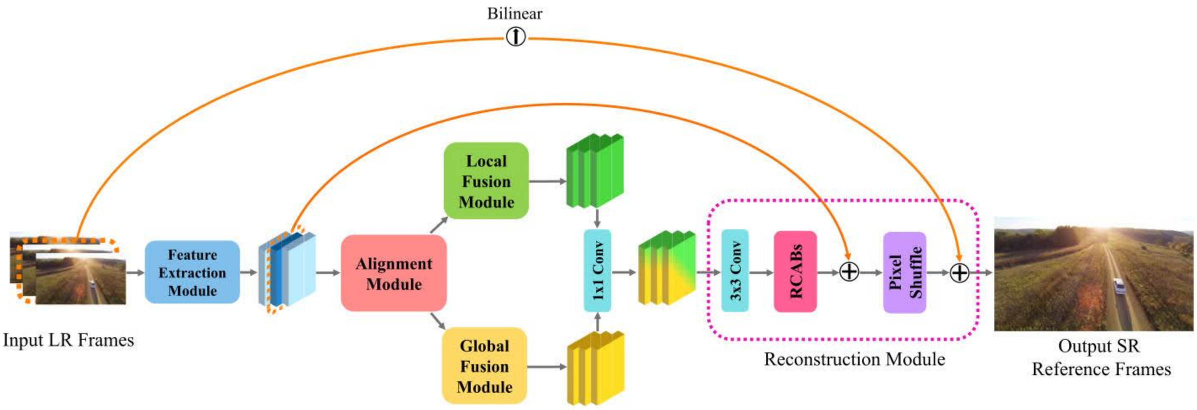
A combination of an RNN-based optical flow method that preserves spatio-temporal information in the current and adjacent frames as the generator and a discriminator that is adept at ensuring the generated SR frame offers superior fidelity.



Added to the benchmark by MSU G&M Lab
Links: GitHub, paper

C.5. LGFN

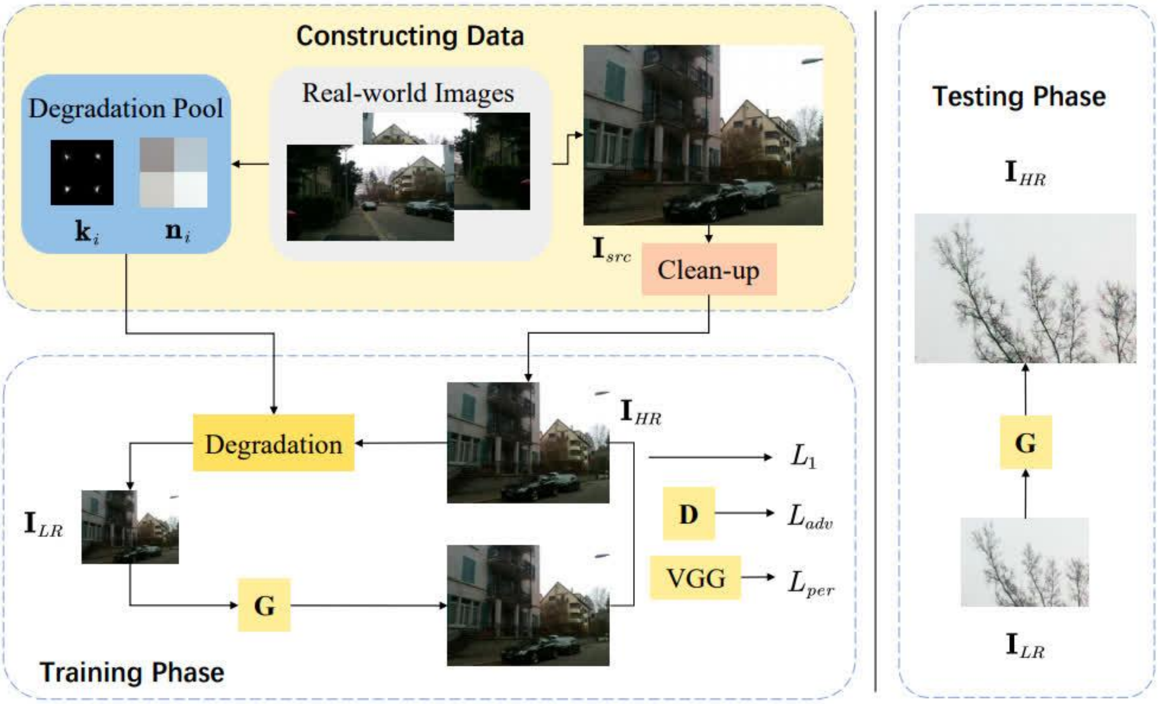
Uses deformable convolutions with decreased multi-dilation convolution units (DMDCUs) to align frames explicitly. Fuses features from local and global fusion modules.



Added to the benchmark by MSU G&M Lab
Links: GitHub, paper

C.6. RealSR

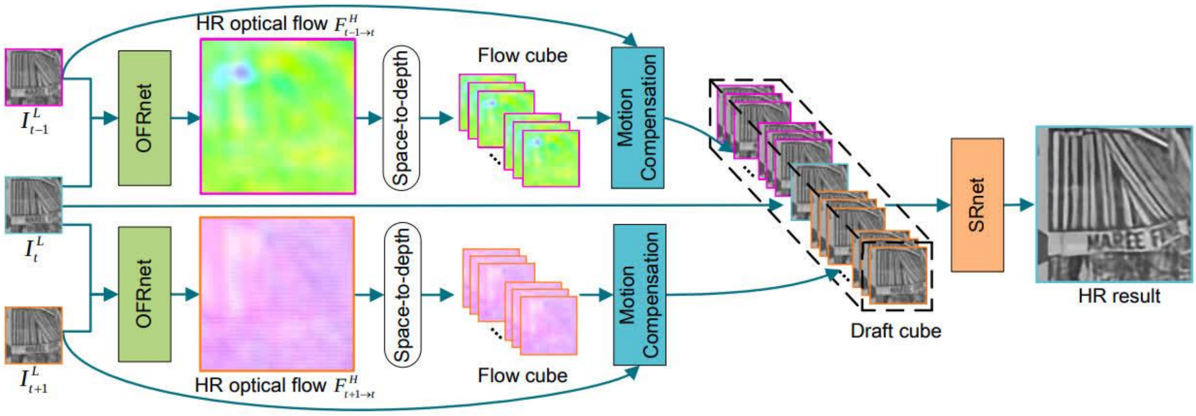
Tries to estimate degradation kernel and noise distribution for better visual quality.



Added to the benchmark by MSU G&M Lab
Links: GitHub, GitHub, paper

C.7. SOF-VSR

Two models: SOF-VSR-BD (trained on gauss degradation type), SOF-VSR-BI (trained on bicubic degradation type)
Compensates motion by high-resolution optical flow, estimated from the low-resolution one in a coarse-to-fine manner.



Added to the benchmark by MSU G&M Lab
Links: GitHub, paper

C.8. Topaz Video Enhance AI

Topaz Video Enhance AI is a commercial filter.

- Three models: ahq-11 (Artemis High Quality v11) — upscale or sharpen high quality input video, reducing motion flicker,
- amq-12 (Artemis Medium Quality v12) — upscale or enhance medium quality video with moderate noise or compression artifacts,
- amqs-1 (Artemis Dehalo v1) — upscale or enhance medium quality progressive video that contains haloing, moderate noise or compression artifacts.

Added to the benchmark by MSU G&M Lab
Links: Website

C.9. waifu2x-ncnn-vulkan

Two models: waifu2x-anime and waifu2x-cunet

NCNN implementation of waifu2x converter. Runs fast on Intel / AMD / Nvidia with Vulkan API. waifu2x-ncnn-vulkan uses the NCNN project as the universal neural network inference framework.

Added to the benchmark by MSU G&M Lab
Links: GitHub

D. OBJECTIVE-QUALITY METRIC DESCRIPTION

D.1. PSNR

PSNR is a commonly used metric for reconstruction quality for images and video. In our benchmark, we calculate PSNR on the Y component in YUV colorspace.

Since some Super-Resolution models can generate images with a global shift relative to GT, we calculate shifted PSNR. We check each shift in the range [-3, 3] (including subpixel shifts) for both axes and select the highest PSNR value among these shifts. We noticed that SRs' results on the same video decoded with different bitrates usually have the same global shift. Thus we calculate the best shift only once for each video.

For metric calculation, we use the MSU Video Quality Measurement Tool (VQMT).

D.2. MS-SSIM

SSIM is a metric based on structural similarity. In our benchmark, we use Multiscale SSIM (MS-SSIM), which is conducted over multiple scales through a process of multiple stages of sub-sampling. We calculate MS-SSIM on all 3 components in the YUV colorspace, and the metric result is normalized[4] the following way: (4Y + U + V) / 6, where Y, U, and V are the MS-SSIM values on Y, U, and V components respectively.

MS-SSIM results also rely on the shift of frames. We take the optimal subpixel shift for PSNR and apply it to input frames before calculating MS-SSIM.

For metric calculation, we use the MSU Video Quality Measurement Tool (VQMT).

D.3. Y-VMAF and Y-VMAF NEG

Y-VMAF is a perceptual video quality assessment algorithm developed by Netflix[3]. We use both Y-VMAF and Y-VMAF NEG (no enhancement gain) in our benchmark.

For metric calculation, we use the MSU Video Quality Measurement Tool (VQMT). For Y-VMAF we use -set "disable_clip=True" option of the MSU VQMT.

It was experimentally proven that shifted Y-VMAF and shifted Y-VMAF NEG give less than 1% gain relative to unshifted versions, that's why we use unshifted versions in our benchmark.

---

[4] A. Antsiferova, A. Yakovenko, N. Safonov, D. Kulikov, A. Gushin, D.Vatolin, "Objective video quality metrics application to video codecs comparisons: choosing the best for subjective quality estimation," *arXiv preprint*, 2021

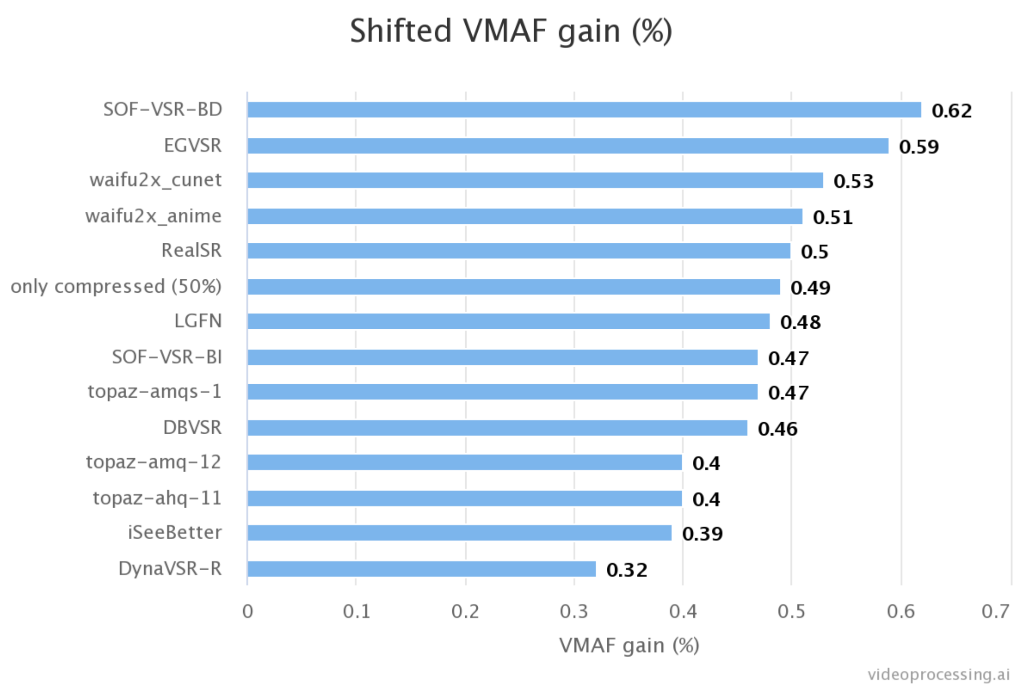


Figure 71a: Average shifted Y-VMAF gain relative to the unshifted version.

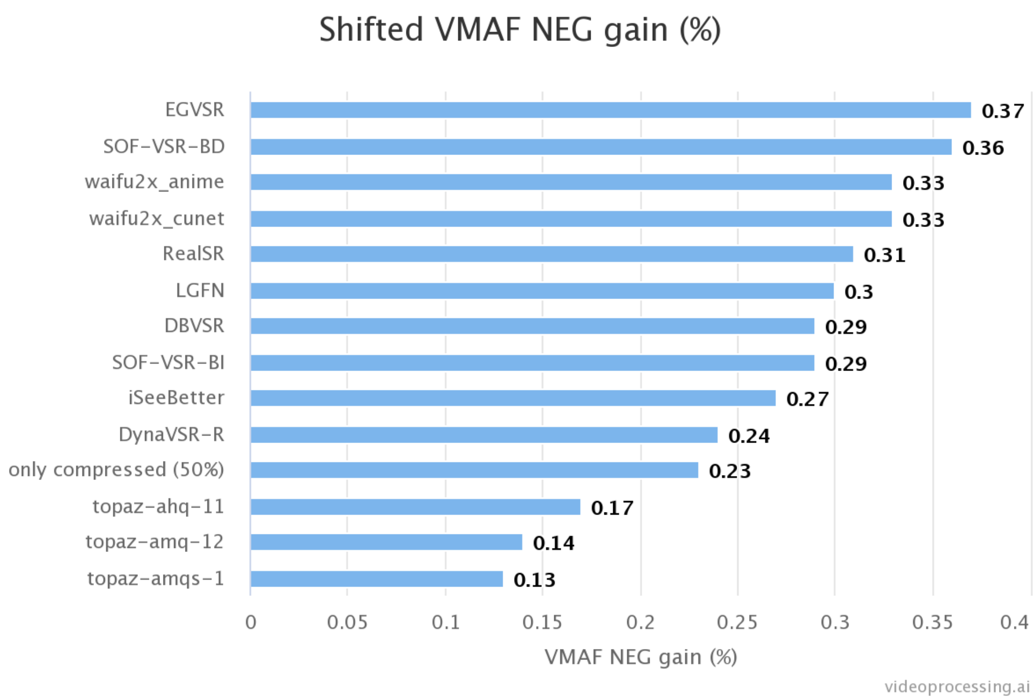


Figure 71b: Average shifted Y-VMAF NEG gain relative to the unshifted version.

The MSU Video Quality Measurement Tool can be downloaded or purchased at http://compression.ru/video/quality_measure/vqmt_download.html#start.

D.4. LPIPS

LPIPS (Learned Perceptual Image Patch Similarity) evaluates the distance between image patches. Higher means further/more different. Lower means more similar. In our benchmark, we subtract the LPIPS value from 1. Thus, more similar images have higher metric values. To calculate LPIPS we use the Perceptual Similarity Metric implementation proposed in The Unreasonable Effectiveness of Deep Features as a Perceptual Metric[5].

It was also proven that shifted LPIPS give less than 1% gain relative to the unshifted version. That's why we calculate LPIPS without shift compensation in our benchmark.

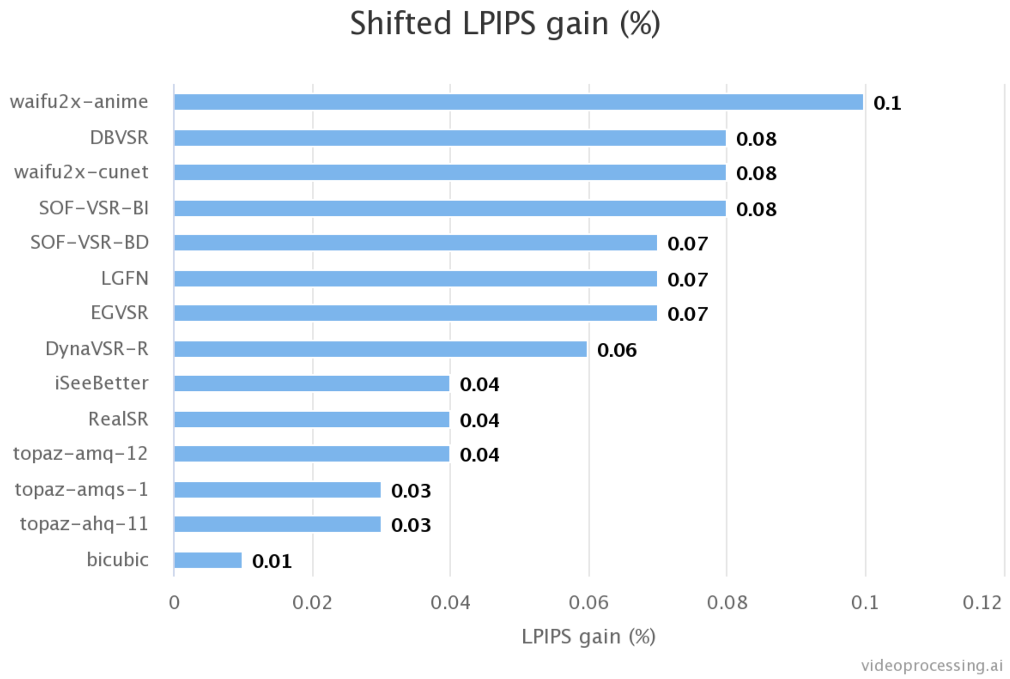


Figure 72: Average shifted LPIPS gain relative to the unshifted version.

D.5. ERQAv1.0

ERQAv1.0 (Edge Restoration Quality Assessment, version 1.0) estimates how well a model has restored edges of the high-resolution frame. This metric was developed for MSU Video Super-Resolution Benchmark 2021.
Firstly, we find edges in both output and GT frames. To do it we use OpenCV implementation of the Canny algorithm. A threshold for the initial finding of strong edges is set to 200 and a threshold for edge linking is set to 100. Then we compare these edges by using an F1-score. To compensate for the one-pixel shift, edges that are no more than one pixel away from the GT's are considered true-positive.
More information about this metric can be found at the Evaluation Methodology of MSU Video Super-Resolution Benchmark.

[5]R. Zhang, P. Isola, A. A. Efros, E. Shechtman, O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2018.

Output frame ERQAv1.0 visualization GT

Figure 73: ERQAv1.0 visualization
White pixels are True Positive, red pixels are False Positive, blue pixels are False Negative

E. FIGURE EXPLANATION

E.1. Rate-Distortion Curves

The Rate-Distortion charts show variation in codec quality by bitrate. For this metric, a higher value presumably indicates better quality.

E.2. Bitrate Ratio for the Same Quality

The first step in computing the average bitrate ratio for a fixed quality is to invert the axes of the bitrate/quality graph (see Figure 68). All further computations use the inverted graph.

The second step involves averaging the interval over which the quality axis is chosen. The averaging is only over those segments for which both models yield results. This limitation is due to the difficulty of developing extrapolation methods for classic RD curves; nevertheless, even linear methods are acceptable when interpolating RD curves.

The final step is the calculation of the area under the curves in the chosen interpolation segment and the determination of their ratio (see Figure 74). This result is an average bitrate ratio at a fixed quality for the two models. When considering more than two models, one of them is defined as a reference model (in our case it's “only compressed”), and the quality of the others is compared with that of the reference.

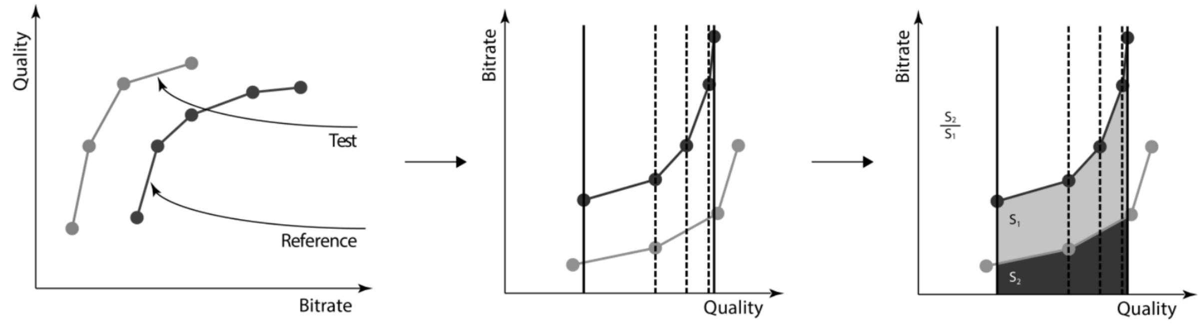


Figure 74: BSQ-rate evaluation

References

1. https://videoprocessing.ai/benchmarks/super-resolution-for-video-compression.html
2. https://compression.ru/video/codec_comparison/2021/
3. http://compression.ru/video/quality_measure/video_measurement_tool.html
4. https://videoprocessing.ai/index.html
5. http://app.subjectify.us/
6. https://en.wikipedia.org/wiki/Bradley%E2%80%93Terry_model
7. Y. Wang, S. Inguva, B. Adsumilli, “YouTube UGC dataset for video compression research,” in *IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)* (pp. 1-5), 2019
8. A. Zvezdakova, D. Kulikov, S. Zvezdakov, D. Vatolin, "BSQ-rate: a new approach for video-codec performance comparison and drawbacks of current solutions," in *Programming and computer software*, vol. 46, 2020, pp.183-194.
9. A. Antsiferova, A. Yakovenko, N. Safonov, D. Kulikov, A. Gushin, D.Vatolin, "Objective video quality metrics application to video codecs comparisons: choosing the best for subjective quality estimation," *arXiv preprint* arXiv:2107.10220, 2021
10. R. Zhang, P. Isola, A. Efros, E. Shechtman, O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586-595, 2018