LAVIB: A Large-scale Video Interpolation Benchmark

Alexandros Stergiou
University of Twente, NL
a.g.stergiou@utwente.nl

Abstract

This paper introduces a **LA**rge-scale **V**ideo **I**nterpolation **B**enchmark (LAVIB) for the low-level video task of Video Frame Interpolation (VFI). LAVIB comprises a large collection of high-resolution videos sourced from the web through an automated pipeline with minimal requirements for human verification. Metrics are computed for each video's motion magnitudes, luminance conditions, frame sharpness, and contrast. The collection of videos and the creation of quantitative challenges based on these metrics are under-explored by current low-level video task datasets. In total, LAVIB includes 283K clips from 17K ultra-HD videos, covering 77.6 hours. Benchmark train, val, and test sets maintain similar video metric distributions. Further splits are also created for out-of-distribution (OOD) challenges, with train and test splits including videos of dissimilar attributes. [1]

1 Introduction

Long uncompressed video streams capture events over varying motion intensities, light conditions, and color dynamic ranges. Although loading and storing individual videos is rudimentary, processing and reading large volumes can bottleneck availability. The high-volume transfer of videos with large filesizes can also result in bandwidth overheads and long decoding times. Low-level vision tasks such as Video Frame Interpolation (VFI) [3, 10, 16, 19, 21, 30, 41, 42, 44, 50, 75], Video Super-Resolution (VSR) [5, 13, 17, 18, 22, 28, 31, 54, 63, 65], and Video Denoising (VD) [14, 32, 33, 53, 58, 61, 64] aim to address such challenges by enabling the storage and stream of lower-resolution, lower-frame-rate, compressed videos. Despite the wide application of such approaches to adjacent tasks such as localization and mapping [26, 71], object tracking [74], novel view synthesis [43, 52], and slow-motion video generation [19, 20], existing datasets for low-level video tasks [2, 40, 41, 46, 55, 56, 59–61, 72] contain short videos, with a small number of frames per video. With the exception of [72] most of these datasets only include either a few hundreds [2, 41, 46, 61] or thousands [40, 55, 56, 59, 60] of videos with limited variations in the motions, luminance, and object-level sharpness. To address this gap, this paper introduces a **LA**rge **V**ideo **I**nterpolation **B**enchmark (**LAVIB**), for learning to interpolate high-resolution videos across varying motion, blur, luminance, and contrast settings. LAVIB is built on per-frame metrics that quantitatively measure motion magnitudes, frame sharpness, video contrast, and overall luminance. In Fig. 1, LAVIB videos are visualized over axes corresponding to the metrics used.

The selected metrics establish a diverse, general, and robust benchmark for VFI as most prior efforts have focused on specific settings. Seminal works [38, 55] sourced videos from high frame-rate sensors that are less relevant to videos recorded by commonly used devices. Other works use videos of standardized resolutions and frame rates. These are either datasets of larger sizes with low-resolution videos [59, 72] or smaller datasets of high-resolution [41, 55, 56, 60, 61]. Comparisons to other video datasets across metrics are discussed in §2.

LAVIB contains 283,484 video segments totaling approximately 77.6 hours. The segments are sourced from 17,204 clips with 3840×2160 (4K) resolution and 60 frames-per-second (fps). Statistics are

---

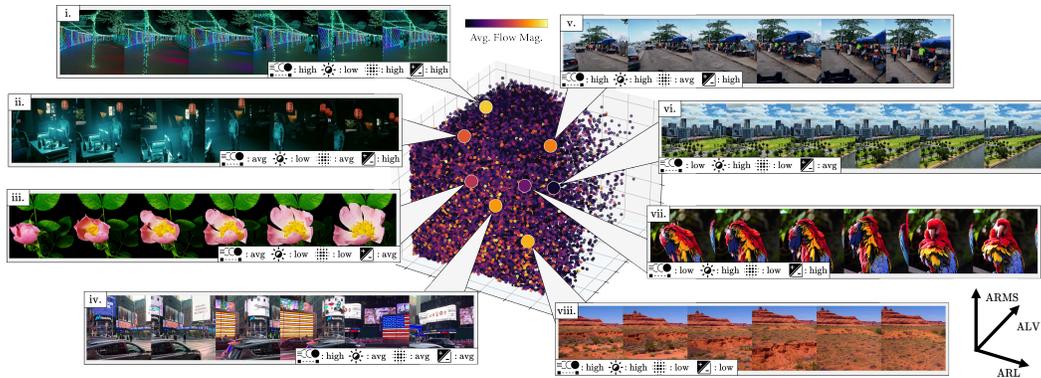[1]LAVIB is accessible at https://alexandrosstergiou.github.io/datasets/LAVIB.

Figure 1: **LAVIB videos distributed across metrics**. Four metrics are computed per video. Average Flow Magnitude (AFM) quantifies motion. The Average Laplacian Variance (ALV) is used to describe the sharpness of frames. The Average Root Mean Square (ARMS) is used for contrast. The Average Relevant Luminance (ARL) relates to the video brightness. The four aforementioned metrics are used for Out-Of-Distribution (ODD) challenges: Fast → slow. and slow. → fast motions. Low → high and high → low sharpness. Low → high and high → low contrast. Bright → dark and dark → bright luminance.

discussed in §3. Similar to previous efforts comprised of 4K videos [38, 55, 60], LAVIB is compiled by temporally and spatially cropping tubelets from the 4K videos to fit clips into memory.

The data collection pipeline for LAVIB is detailed in §4. This includes the creation of a vocabulary of search query terms. Clips are sourced from YouTube videos queried by search terms. Preset clip sampling intervals are used to standardize durations. Segments are selected from high average flow magnitude temporal locations calculated with [15]. Spatial locations are selected by tubelets of high/low metrics values. The final train/val/test sets are constructed by balancing all metrics.

Widely-used VFI methods [16, 23, 75] are benchmarked on the LAVIB val and test sets in §5. Performance is reported across well-adopted evaluation metrics [4, 8, 9, 12, 29, 76]. LAVIB's large size and video diversity enables pretraining models of greater generalizability that are in turn evaluated on test sets of smaller down-stream datasets targeting either scene diversity [72], high frame rates [55], or high video resolution [38, 41]. In addition to the main benchmark splits, four challenges with two settings each, are introduced for Out-Of-Distribution (OOD) VFI. Train, val, and test sets with unbalanced metric distributions are created for each challenge and setting. Videos are assigned to sets based on their average motion magnitude, sharpness, contrast, and luminance metrics. These challenges evaluate model generalizability over diverse domains that are different in the train and test sets.

2 Related works

Initial VFI benchmarks [2] provided real image sequences and ground truth optical flow annotations with average resolutions of 640×480. The dataset comprised a small number of videos used primarily for evaluation. Vid4 [34] is a standardized testing benchmark for VFI and VSR consisting of four videos of 740×480 and 720×576 resolutions. Similarly, [73] is also used for VSR with videos sampled from [68]. Later efforts [39] have also introduced benchmarks for VD in tandem with VSR and VFI. More recent works [40] included 3.2K HD videos captured with a GOPRO4 Hero Black with frame averaging to simulate lower shutter speeds. [51] also proposed a synthetic dataset with 3D objects from [6] and backgrounds from [24, 27]. The trajectories of objects were uniformly sampled from fixed bounds. Works have also studied VFI for specific domains such as animations [56]. [7] introduced benchmarks under large motion conditions with 20 240-fps videos sourced from YouTube. Recently, [55] introduced a high-resolution high-frame rate benchmark for video interpolation and super-resolution. It includes a total of 4,423 videos recorded with a Phantom Flex4K.

Most similar to LAVIB, adjacent efforts that compile 4K video datasets [38, 41, 55, 60] source videos from media in which professional equipment are used; e.g. movies [38, 41] or high-resolution video recordings [55]. Videos from these datasets are primarily recorded with sensors under optimal shutter speeds and calibrated luminance for capturing specific motion types. In contrast, LAVIB includes videos from various sensors such as hand-held, action, professional, or drone cameras, and screen

Table 1: **Datasets**. Compared to prior efforts, LAVIB provides a large-scale general-purpose dataset of standardized 4K 60 fps videos. It features a significant variance across Average Flow Magnitude (AFM), Average Relevant Luminance (ARL), and Average Laplacian Variances (ALV) in videos.

Dataset	Dataset statistics			Video statistics		Average video metrics		
	Year	Tot. Mins	Tot. Vids	Src Res.	FPS	AFM	ARL	ALV
UCF101 [59]	2012	1,600	13,320	240p	25	2.43 ± 1.85	53.37 ± 13.42	53.99 ± 18.37
Xiph [38, 41]	2020	4	19	**2160p**	60	26.21 ± 25.19	60.64 ± 10.77	95.24 ± 62.32
Inter4K [60]	2021	83	1,000	**2160p**	60	56.38 ± 14.34	56.79 ± 14.48	25.05 ± 24.05
X4K1KFPS [55]	2021	191	4,423	**2160p**	960	266.87 ± 178.72	53.95 ± 12.07	135.67 ± 78.19
Vimeo90K [72]	2017	356	91,701	720p	30	49.63 ± 18.32	59.68 ± 20.89	26.26 ± 29.25
LAVIB (ours)	2024	**4,660**	**283,484**	**2160p**	60	63.10 ± 58.41	38.34 ± 28.69	199.78 ± 197.79

Table 2: **LAVIB split statistics**. Details per metric for each split.

	Statistic	Train	Val	Train+Val	Test
⠿	# Low Flow Mag	19,605 (10.3%)	3,846 (9.3%)	23,451 (10.1%)	4,898 (9.1%)
	# High Flow Mag	18,976 (10.1%)	3,891 (9.4%)	22,867 (9.9%)	5,482 (10.2%)
⠿	# Low Lap. Var.	18,313 (9.6%)	3,541 (8.6%)	21,854 (9.5%)	6,494 (12.1%)
	# High Lap. Var.	17,348 (9.2%)	3,871 (9.4%)	21,219 (9.2%)	7,130 (13.3%)
☼	# Low Perc. Lum.	17,669 (9.3%)	3,638 (8.8%)	21,307 (9.2%)	7,041 (13.1%)
	# High Perc. Lum.	19,297 (10.2%)	4,400 (10.7%)	23,697 (10.3 %)	4,652 (8.6%)
◩	# Low RMS Cont.	18,794 (10.0%)	3,657 (8.8%)	22,451 (9.8%)	5,897 (11.0%)
	# High RMS Cont.	18,363 (9.7%)	4,036 (9.8 %)	22,399 (9.7%)	5,950 (11.1%)
	Total	188,644	41,345	229,989	53,494

captures. The videos differ in their dynamic range, levels of post-processing, and compression. LAVIB is intended as a general-purpose dataset and benchmark without being specific to sensor types or settings. Examples of videos are shown in Fig. 1.

LAVIB is compared in Tab. 1 to adjacent video datasets over different statistics. **Dataset statistics** include the number of videos and total running times. **Video statistics** relate to video information such as the resolution and frame rate. **Average video metrics** provide metrics on the variance of motions, lighting conditions, and frame sharpness. Definitions of the metrics are detailed in §3. LAVIB has threefold more videos than [72] and equally larger total video running time than [59]. The difference in LAVIB video conditions and recording sensors is reflected by the high variance across metrics in Tab. 1. With the exception of [55], tailored for videos of fast motions with high optical flow magnitude, LAVIB has the highest variance per metric across datasets.

3 LAVIB statistics

Four statistics are used to obtain segments, create splits, and define challenges. An overview is shown in Tab. 2 with the number of videos with the highest/lowest metrics reported.

Frame-pair motion. A significant challenge for VFI methods is learning to model the cross-frame motion consistency of videos. Thus, the proposed dataset includes videos of diverse magnitudes; both high camera or object motion, and more static scenes. Motion magnitudes can be quantified with dense optical flow. FlowFormer [15] is used on each frame pair resulting in 598 frame pairs per video. The spatial resolution of videos is reduced by ×0.25 to fit frames in memory. The Averaged Flow Magnitude (AFM) is defined by spatio-temporally averaging optical flow. AFM variances are reported for all datasets in Tab. 1.

Frame sharpness. Sourced videos vary by the sensors, lens, codex, and camera profiles used. They can capture different motions, light conditions, and camera focus. All these factors amount to significant variations in the sharpness of videos. Thus, object edges or sensory noise may be highlighted or suppressed. The Laplacian of Gaussians (LoG) is a standardized kernel-based approach for highlighting regions of rapid change in pixel intensities. Given a video $\mathbf{V}$ of dimensions $\mathbb{R}^{D=T \times H \times W}$, with T frames, H height, and W width, it convolves a kernel with size K over each frame. ALV is formulated by applying LoG and averaging:

$$\text{ALV}(\mathbf{V}, \sigma, K) = \frac{1}{D} \sum_{r \in \mathbb{R}^D} \sum_{i=1}^{K} \sum_{j=1}^{K} \underbrace{-1 \frac{1}{\pi \sigma^4} (1 - \frac{i^2 + j^2}{2\sigma^2}) e^{-\frac{i^2 + j^2}{2\sigma^2}}}_{\text{LoG}(i,j) \text{ kernel}} \mathbf{V}_{r - [i,j]} \qquad (1)$$

As the size of the kernel also factors the estimate, an ensemble of kernel sizes $\mathcal{N} = \{3, 5, 7\}$ is used to calculate the final value $\frac{1}{|\mathcal{N}|} \sum_{K \in \mathcal{N}} \text{ALV}(\mathbf{V}, \sigma, K)$ with $\sigma = 1.4$. Overall, in LAVIB 18,313 train, 3,541 val, and 6,494 test videos are at the upper 10% of the ALV ensemble.

Video contrast. Another characteristic of videos is the contrast between objects and backgrounds in scenes. The human visual system is more sensitive to the contrast between foreground and background [37, 49], compared to other adjacent measures such as the perceived luminance (brightness), or frame sharpness (blurriness). Computationally, contrast relates to the difference between neighboring raw pixel values. The metric is formulated as the *Average Root Mean Square* (ARMS) [45] difference between each pixel from each frame of $\mathbf{V}$ and the corresponding pixel in the channel-averaged $\overline{\mathbf{V}}$.

$$\text{ARMS}(\mathbf{V}) = \frac{1}{T} \sum_{t \in \mathbb{R}^T} \sqrt{\frac{1}{HW} \sum_{s \in \mathbb{R}^{HW}} (\mathbf{V}_{t,s} - \overline{\mathbf{V}}_{t,s})} \qquad (2)$$

LAVIB includes 22,399 videos of high contrast for train and val and, 22,451 videos of low contrast.

Luminance conditions. In addition to the overall video conditions, the perception of light can be affected by the sensor's sensitivity or the camera's processing. In human vision, the perception of luminance is done over three bands of color. To account for the uneven perception of each band, a common standard for quantitatively defining luminosity is the relevant luminance [47]. In videos, the Average Relative Luminance (ARL) can be computed as the weighted sum for each color channel from video frames based on [47], which in turn is averaged over time. The bottom 10th ARL percentile in LAVIB includes 17,669 train, 3,638 val, and 7,041 test videos. Similarly, there are 19,297 train, 4,400 val, and 4,652 test high-luminance videos.

As shown in Tab. 2, videos selected for all splits are balanced across metrics. This is done explicitly for the main benchmark and not the OOD challenges.

4 LAVIB pipeline

The video selection pipeline includes several stages for the collection, extraction, and set assignment. Initially, videos are searched on YouTube by textual prompts designed to return relevant videos with 4K resolutions and 60 frames per second as overviewed in §4.1. Sourced videos are cropped to 10-second clips standardizing their durations and improving processing speeds in further steps of the collection pipeline. Segments with high motion magnitudes are selected from the clips and are cropped to tubelets by their AVL, ARL, ARMS, and AFM statistics as detailed in §4.2. Dataset splits are balanced between the four statistics, with OOD splits created by assigning videos with the highest average metric at the test or train set. The dataset pipeline is flexible and can be scaled over large numbers of videos, requiring manual input only at a few points.

4.1 Video web-crawling

The first stage of the data collection pipeline constructs queries to search and identify videos on YouTube with 4K resolution and 60 fps. The vocabulary of search terms is created from a finite combination of different categories e.g.; locations, activities, weather conditions, and camera types. This aims to diversify results over the defined categories with a level of control (See appendix A1 for a full discussion on vocabulary creation). The vocabulary terms are compiled with three guidelines.

Videos should be in the wild. Retrieved videos should vary by lighting conditions, motions, and scenes. They should also be recorded with different sensors. Sensor types depend strongly on video themes; e.g. action cameras are more common for capturing fast-paced scenes in contrast to DSLR cameras. Conditions are added in the format; `rainy walk in New York` or `night drive`. Some text prompts are also designed to include specific equipment such as `GoPro Hero10` or `iPhone 13 Pro`.
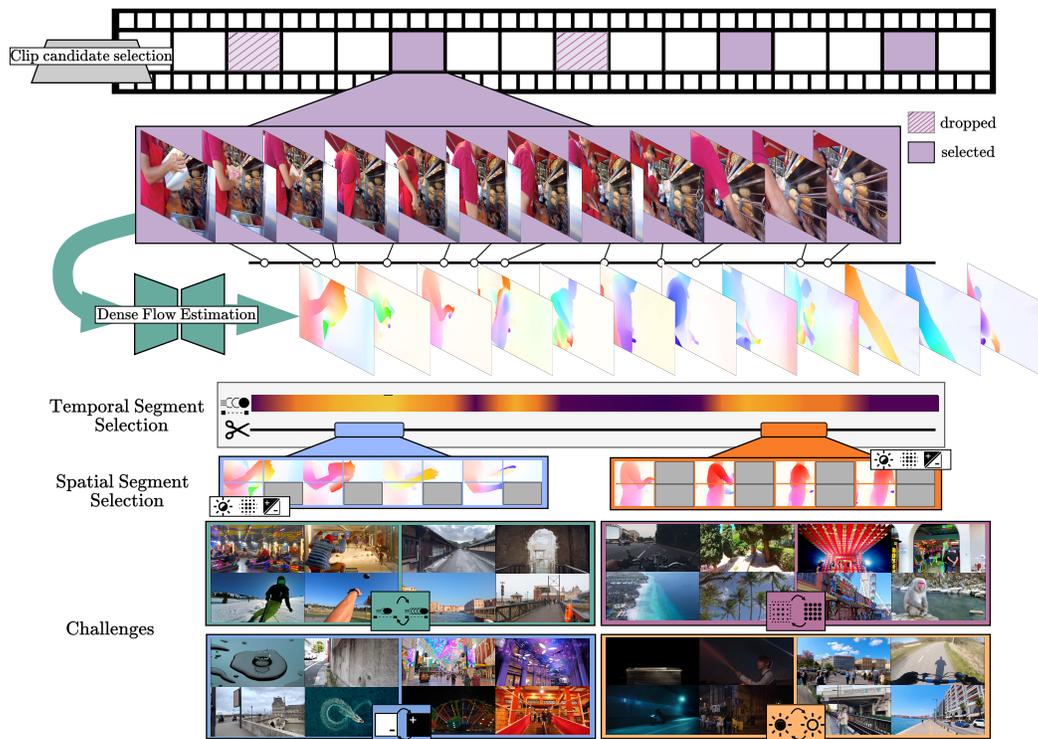
Figure 2: **LAVIB segment selection and challenges pipeline**. Candidate 10-second clips are sampled from a long video based on their embedding similarity. Dense optical flow is computed with [15] and spatially averaged for the AFM metric. The 1-second clips with the top-20% AFM are selected for the next step. Clips are further partitioned into four tubelets used in the final dataset based on their ARL, ALV, ARMS, and AFM. The metrics are also used for video selection in OOD challenges for a. motion, b. sharpness, c. contrast, d. luminance.

Video content should correspond to raw footage. The dictionary of general search terms aims to improve control over the video context by retrieving specific video types. Videos with substantial post-production cuts, or transitions, can be less relevant or usable for VFI. The video types that are collected focus primarily on raw footage.

Exclusivity of video categories. Vocabulary queries should also include diversity in the themes present. This is done by constructing verb hierarchies. A balanced number of queries is constructed for objects/locations that are the focus of the videos.

Each vocabulary search term is combined with '4K' and used as a query on YouTube. The query-related URLs are scraped from the contents in the response's script. Candidate videos are downloaded only if a 4K format with 60 fps is available. This step is needed as YouTube's search prioritizes video elements such as titles, tags, and descriptions over metadata.

Limitations. Queries are created from a finite set of search terms. The diversity of locations and activities is manually defined thus, limitations are expected. As noted above, the selected videos are more diverse than current VFI benchmarks however, an increased vocabulary can improve this further.

4.2 Segment selection and split assignment

In total, 667 hours of footage are collected over the project's 31-month duration. This initial list contained videos of hour-long to minute-long durations. To standardize their durations, 10-second clips are sampled manually over different interval steps. Clips are extracted consecutively for videos less than 5 minutes. For the rest of the videos; 10-second sampling intervals are used for videos with durations between 5-30 minutes, 2-minute intervals for videos between 30 minutes to an hour, and 10-minute intervals for videos longer than an hour. This selection resulted in a total of 34,408 clips. Clips from the same video are bound to include similarities. To account for this and inspired by [76], similarities between clips from the same video are measured metrically by their embedding space distance with highly similar clips being dropped. MViTv2-B [11] is used to encode clips and

Table 3: **LAVIB val and test results** using [23] as a baseline across training schemes. Evaluation metrics are reported for both val and test sets. Best results per metric are denoted in **bold**.

Pre-train LAVIB	Pre-train Vimeo-90K	Fine-tune Xiph + X4K1KFPS	LAVIB val performance			LAVIB test performance		
			PSNR↑	SSIM↑	LLIPS↓	PSNR↑	SSIM↑	LLIPS↓
	✓		32.86	0.968	$3.152e^{-2}$	32.10	0.963	$3.947e^{-2}$
	✓	✓	31.36	0.952	$4.620e^{-2}$	31.78	0.948	$5.154e^{-2}$
✓			**33.72**	**0.981**	**$2.515e^{-2}$**	**33.44**	**0.981**	**$2.934e^{-2}$**

to create a similarity matrix based on the L2 distance of the final layer embeddings. Clips with an average (row-wise) L2 distance below the entire matrix's average distance are dropped. This step resulted in the selection of 17,204 clips.

The final two stages include both temporal and spatial cropping. They are overviewed in Fig. 2.

Temporal segment selection. Segments are compared and selected by their AFM. This selection aims to drop primarily static segments as they are less relevant to VFI tasks with minimal pixel and object tracking requirements. FlowFormer [15] is used to calculate AFM over pairs of frames by spatially averaging flows. Each 10-second sequence is temporally augmented to obtain all available 1-second clips. Clips with the highest 20% magnitudes are selected. This strategy was chosen as it worked well in a small-scale setting when manually examining a set of 1,000 clips.

Spatial segment selection. The selected high-resolution clips cannot directly fit into the memory of most current GPUs. Thus, as commonly addressed in the literature [38, 41, 55, 72] the number of videos is curated with the additional selection of tubelets. Each clip is divided into four tubelets by a 2×2 grid. ALV, ARL, ARMS, and AFM are computed for each tubelet. 80% of the tubelets are retained by selecting from the low/high values per metric in succession, leaving out the middle 20%. This avoids oversampling from values close to the mean of metrics. Instead, tubelets with more challenging settings are selected.

Assignment to splits. All train/val/test splits are constructed with a 65-15-20% split. DUPLEX [57] selection is used for balancing split statistics. Videos with the largest pair-wise distance by their metrics are initially selected. In turn, videos are iteratively assigned to sets given their distance from the previously selected videos. A detailed overview of the algorithm is provided in §A2. Recall that the OOD sets need to be imbalanced across statistics so this is specific to the benchmark splits.

Limitations. No prior work has tackled video collection based on these metrics, so thresholds for each step are manually defined. This can constrain the final dataset size as the values were selected empirically to maximize diversity.

5 Benchmarks

Baselines. LAVIB contains 188,644 1-second videos for training, 41,345 videos for validation, and 53,494 videos for testing. Benchmark results are reported in §5.1 across settings. For the baselines, triplets of frames are defined similarly to [7, 59, 72] for single-frame interpolation with a total of ~5.7M triplets. In the multi-frame interpolation settings in §5.2, septuplets are also used resulting in a total of ~ 2.4M groups of frames. Ablations on varying video resolutions are presented in §5.3. In §5.4, the video metrics are used to create *unbalanced* dataset splits. For each of the four metrics, two challenges are created by sampling videos with either high/low values and assigning them to the train/test sets. Qualitative results for all three models are shown in §5.5.

Model details. Three VFI methods are benchmarked; RIFE [16], EMA-VFI [75], and FLAVR [23], which in turn are trained and tested on LAVIB. The official codebases made publicly available by their respective authors are adjusted and used for LAVIB for all experiments. Adapted training and test code, and models are available at https://github.com/alexandrosstergiou/LAVIB.

Training details. The training and model settings are imported from the original papers and codebases. The train batch size is set to 64 for all models and the start learning rate is reduced by $\times 0.25$ for all models to account for the increased batch size.

Evaluation metrics. Standard image and video quality metrics are used for all tasks and benchmarks. Quantitative results report the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) [76]. In multi-frame interpolation, the average value over the interpolated frames is reported.

Table 4: **Vimeo-90K performance** [23] with dif. train sets.

Train set	PSNR↑	SSIM↑	LPIPS↓
Vimeo-90K	36.25	0.975	$9.280e^{-3*}$
LAVIB	**36.68**	**0.983**	**$4.162e^{-3}$**

Table 5: **Xiph4K performance** [23] with dif. train sets.

Train set	PSNR↑	SSIM↑	LPIPS↓
Vimeo-90K	33.28^*	0.892^*	$0.236e^{-1*}$
LAVIB	**34.51**	**0.911**	**$0.532e^{-2}$**

Table 6: **X4K1KFPS performance** [23] with dif. train sets.

Train set	PSNR↑	SSIM↑	LPIPS↓
Vimeo-90K	31.25^*	0.9083^*	$0.383e^{-1*}$
LAVIB	**32.44**	**0.927**	**$0.894e^{-2}$**

Table 7: **Multi-metric evaluation results on LAVIB test**. Performance is reported for ⋆image-based metrics averaged across frames and ♦video-based metrics.

Model	⋆PSNR↑	⋆SSIM↑	⋆LPIPS↓ [76]	⋆DISTS↓ [9]	⋆Watson-DFT↑ [8]	♦VSFA↑ [29]	♦VFIPS↑ [12]
RIFE	27.88	0.871	$1.416e^{-1}$	$1.870e^{-1}$	0.215	0.558	0.561
EMA-VFI	33.14	0.978	$3.105e^{-2}$	$5.076e^{-2}$	0.344	0.607	0.638
FLAVR	**33.44**	**0.981**	**$2.934e^{-2}$**	**$4.430e^{-2}$**	**0.360**	**0.626**	**0.667**

5.1 Baseline results

Baselines. Tab. 3 reports SSIM, PSNR, and LPIPS scores on both LAVIB val and test sets across three training settings; pre-training on Vimeo-90K, fine-tuning on a joint set from Xiph [38, 41] and X4K1KFPS [55] of exclusively 4K videos, and pre-training with LAVIB. FLAVR [23] is used as the baseline model due to its fast processing times, strong results, and open-source codebase. Finetuning on Xiph + X4K1KFPS suffers as both datasets are small in size although they are sourced by videos with the same resolution as LAVIB. Pre-training only on Vimeo-90K slightly improves results. Pre-training on LAVIB gives the best performance overall increasing PSNR, and SSIM by +1.08 and +0.015 on average on both sets.

Generalization to related small-scale datasets. VFI benchmarks include multiple datasets [38, 41, 55, 72]. LAVIB is unique in having the largest number of diverse videos of *both* high resolution and high frame rates. The generalization benefits of using LAVIB as the pre-training dataset are compared to the previously widely-used Vimeo-90K [72]. Tab. 4 shows performance improvements in the test set of Vimeo-90K when the model is trained on LAVIB. Similar score increases are also observed for the Xiph4K and X4K1KFPS test sets in Tabs. 5 and 6 with +1.23 and +1.19 improvements on the PSNR. LAVIB's large variance across videos enables learning VFI over different conditions which can benefit performance in smaller domain-specific benchmark datasets.

Multi-metric results. As human judgment of the perceptual quality depends on high-order image structures and context [36, 67], an ensemble of metrics is reported in Tab. 7 to provide a complete evaluation of each methods' performance on the LAVIB test set. In addition to standard quality metrics, scores over recently-proposed metrics including DISTS [9], Watson-DFT [8], VSFA [29], and VFIPS [12] are also reported. Across statistics, both EMA-VFI and FLAVR perform comparably. A decrease in performance is observed with RIFE as its limited complexity can not adequately address VFI with large variations in settings across videos. Compared to FLAVR, the PSNR and SSIM scores decrease by -5.56 and -1.10 respectively, and the LPIPS loss increases from 0.029 to 0.146.

5.2 Multi-frame interpolation results

This section ablates the number of frames interpolated and evaluated over different schemes with; ×2 interpolation being equivalent to interpolating 30fps videos to 60fps, ×3 interpolating 20fps to 60fps, and ×4 interpolating 15fps to 60fps. Triplets and septuplets of frames as input are also ablated. Results are reported in Tab. 8. FLAVR trained on Vimeo-90K is used as a baseline in all settings.

Varying number of interpolated frames. The LAVIB-trained model [23] consistently outperforms the baseline trained on Vimeo-90K across different numbers of interpolated frames. An average -1.19/-0.02 PSNR/SSIM drop is observed across {×2, ×3, ×4} interpolations when septuplets of frames are used. This drop is more significant for triplets with -1.75/-0.078 PSNR/SSIM.

Varying number of input frames. Two settings are used for defining inputs. In triplets, models input a single proceeding and a single succeeding frame with the interpolation target being the in-between frame. In septuplets, two proceeding and two succeeding frames are used as inputs. Models trained with septuplets demonstrate only moderate PSNR/SSIM performance improvements across interpolation settings. This shows that regardless of the input settings the dataset remains challenging.

---

*Inhouse evaluation from author provided model.

Table 8: **Multi-frame interpolation scores** over triplets, and septuplets across different numbers of interpolated frames. Increase in video duration due to interpolation is denoted with $\{\times 2, \times 3, \times 4\}$.

| Model | triplet | | | septuplets | | |
| | ×2 | ×3 | ×4 | ×2 | ×3 | ×4 |
	PSNR↑/SSIM↓	PSNR↑/SSIM↓	PSNR↑/SSIM↓	PSNR↑/SSIM↓	PSNR↑/SSIM↓	PSNR↑/SSIM↓
Baseline	32.10/0.963	31.58/0.952	30.42/0.937	32.69/0.976	32.10/0.972	31.95/0.918
FLAVR	**33.44/0.981**	**33.07/0.975**	**32.86/0.968**	**33.62/0.985**	**33.41/0.980**	**33.28/0.962**

Table 9: **Results on ×2 interpolation** with different target fps. Main results default settings in gray.

| Model | 15fps → 30fps | | 30fps → 60fps | |
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
FLAVR	33.21	0.978	33.44	0.981

Table 10: **LAVIB test set scores across frame resolutions** with FLAVR on different training schemes. Main results default settings in gray.

| Train set | 112×112 | | | 256×256 | | |
	PSNR↑	SSIM↑	LLIPS↓	PSNR↑	SSIM↑	LLIPS↓
Video90K	30.14	0.943	$4.638e^{-2}$	32.10	0.963	$3.947e^{-2}$
LAVIB	32.57	0.965	$3.781e^{-2}$	**33.44**	**0.981**	**$2.934e^{-2}$**

Table 11: **Frame resolutions ablations**. Best results per metric are denoted in **bold** and best results per model are underlined.

| Model | 112×112 | | | 224×224 | | | 256×256 | | |
	PSNR↑	SSIM↑	LLIPS↓	PSNR↑	SSIM↑	LLIPS↓	PSNR↑	SSIM↑	LLIPS↓
EMA-VFI	32.26	0.954	$4.130e^{-2}$	33.01	0.972	$3.211e^{-2}$	<u>33.14</u>	<u>0.978</u>	$3.105e^{-2}$
FLAVR	32.57	0.965	$3.781e^{-2}$	33.28	0.973	$3.086e^{-2}$	<u>**33.44**</u>	<u>**0.981**</u>	<u>**$2.934e^{-2}$**</u>

Varying frame sampling. LAVIB's standardized 60fps also enables works to explore VFI over more challenging settings with multiple temporal resolutions. Tab. 9 reports performance on 30fps targets created by sampling every 2 frames to form triplets. Results show consistency between densely sampling frames sequentially (30fps → 60fps) and sampling with a step of 2 (15fps → 30fps).

5.3 Varying frame resolution results

An important factor for VFI is the clarity of the objects. Different computational budgets can limit availability in training schemes and memory use.

Resolutions across models. Results on different training set resolutions are reported in Tab. 11. As in [16, 23, 75], 256×256 is the standard resolution used for training all models. A proportional decrease in performance is observed at lower resolutions. However, these reductions remain small with an average $-0.14/-0.01$ in PSNR/SSIM when using 224×224 and $-0.87/-0.02$ when using 112×112. Thus, LAVIB can be a suitable benchmark for evaluating low-compute VFI models.

Frame resolutions across training schemes. Tab. 10 reports performances across varying resolutions with different dataset training sets. Compared to the LAVIB-trained model, performance degrades significantly at lower resolutions with the smaller and less diverse Vimeo-90K. The large and varying LAVIB training set can be an effective alternative for training on lower compute resources in which full-resolution videos do not fit in memory.

5.4 OOD Challenges

OOD challenges aim to test the generalizability of models to domains different from the ones trained. In low to high challenges, train sets include videos of low AFM, ALV, ARMS, or ARL values and the remaining videos of high-value metrics are used for testing. For high to low challenges, train sets have high AFM, ALV, ARMS, or ARL values and test sets have low values.

Low/High AFM. As shown in Tab. 12a, existing VFI models cannot effectively interpolate frames when trained on videos with low motion magnitudes. Compared to the benchmark results in Tab. 7 a -2.64 and -0.04 drop is observed for PSNR and SSIM. The embedding distance to ground truth frames also increases by $+9.825e^{-2}$. In contrast, when models are trained on high motion magnitudes, VFI is easier for the target domain of primarily low magnitudes. The imbalance in performance shows the sensitivity of current models to the motion magnitudes of the training data.

Low/High ALV. Sharpness-based comparisons are reported in Tab. 12b. Testing on low-sharpness settings is more challenging for VFI models as object edges are more difficult to define. However,

Table 12: **PSNR,SSIM, and LPIPS scores on OOD challenges**. Flow-based challenges are denoted by ⬚ →⬚ for low train and high test AFM and ⬚ →⬚ for high train to low test. For blur-based ⬚ →⬚ denotes low and high and ⬚ →⬚ denotes high and low. ⬚ →■ and ■ →⬚ denote low/high, and high/low ARMS respectively. ☼ →☀ and ☀ →☼ denote low/high, and high/low ARL.

(a) AFM

Model	⬚ →⬚			⬚ →⬚		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RIFE	25.34	0.832	$3.816e^{-1}$	28.75	0.926	$8.709e^{-2}$
EMA-VFI	30.21	0.936	$6.420e^{-2}$	34.89	0.929	$1.705e^{-2}$
FLAVR	**30.67**	**0.959**	**$5.094e^{-2}$**	**35.66**	**0.991**	**$1.342e^{-2}$**

(b) ALV

Model	⬚ →⬚			⬚ →⬚		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RIFE	26.52	0.873	$1.823e^{-1}$	29.31	0.906	$9.644e^{-2}$
EMA-VFI	31.26	0.948	$2.947e^{-2}$	34.30	0.972	$2.703e^{-2}$
FLAVR	**31.78**	**0.962**	**$2.942e^{-2}$**	**34.67**	**0.975**	**$2.627e^{-2}$**

(c) ARMS

Model	⬚ →■			■ →⬚		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RIFE	26.28	0.836	$1.766e^{-1}$	25.42	0.855	$2.358e^{-1}$
EMA-VFI	32.79	0.964	$2.930e^{-2}$	30.65	0.951	$4.467e^{-2}$
FLAVR	**33.02**	**0.982**	**$2.561e^{-2}$**	**31.11**	**0.977**	**$3.024e^{-2}$**

(d) ARL

Model	☀ →☼			☼ →☀		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RIFE	26.83	0.872	$1.743e^{-1}$	26.95	0.865	$1.627e^{-1}$
EMA-VFI	33.55	0.974	$2.723e^{-2}$	33.41	0.968	$3.031e^{-2}$
FLAVR	**33.97**	**0.980**	**$2.543e^{-2}$**	**34.20**	**0.976**	**$2.875e^{-2}$**

models trained on low-sharpness videos can interpolate high-sharpness videos with an average +2.93 and +0.023 increase in the PSNR and SSIM scores compared to the low-to-high task.

Low/High ARMS. Results on contrast-based OOD challenges are presented in Tab. 12c. The domain gap between these two settings is significant. Training on low contrast shows robustness when the domain shifts to high contrast at testing. However, the same generalization is not observed for the inverse with models trained on high-contrast videos and tested on low-contrast VFI. Compared to low-to-high ARMS, high-to-low ARMS shows a -1.65 drop in PSNR.

Low/High ARL. Tab. 12d reports performances over brightness settings. Overall, models from either setting show comparable performance and generalization robustness to the target domain. Minor performance improvements are shown for the high to low task with high luminance training being more effective in cross-domain generalization.

5.5 Qualitative results

Fig. 3 shows interpolated frames from the LAVIB test sets. Frame regions from videos of the LAVIB benchmark interpolated with RIFE, EMA-VFI, and FLAVR are shown in the top three row (a-i). Regions shown vary by size and reconstruction error. LAVIB is challenging for current VFI methods as they cannot fully interpolate all parts of objects (b,i) or fine details (c,f,g). Objects in scenes affected by high motions are shown to be the most prone to interpolation artifacts as seen with the fine details being missed (d) and the high cross-frame relative displacement (e). This also becomes apparent more in high-motion scenes (h) where large distortions in the scene dynamics can be observed. For OOD challenges models also struggle to correctly interpolate the high contrast between objects and backgrounds (k,l,n), distinct patterns (j), and details or objects (m,o). Further qualitative results are provided in §A5.

6 Conclusions and future directions

This paper introduces LAVIB, a large-scale general-purpose dataset and benchmark for VFI. LAVIB consists of 283,484 clips collected from 4K videos at 60fps with metrics computed per video specific to motions, sharpness, contrast, and luminance. With the release of the videos and the OOD challenges splits, LAVIB can be used as a robust benchmark and allow the community to investigate VFI under a diverse range of video settings, captured with different equipment, and across various domains.

LAVIB further encourages exploring new avenues for efficiency improvements in future VFI works.

Frame-level quantization. A number of works have explored video inference acceleration through frame quantization for temporal redundancy reduction [1, 62]. Learning to truncate videos by varying quantization precision is important for the real-world applicability of methods in streams. LAVIB provides a diverse set of high-resolution videos with standardized frame rates that can be used both as a benchmark as well as a pre-training dataset.
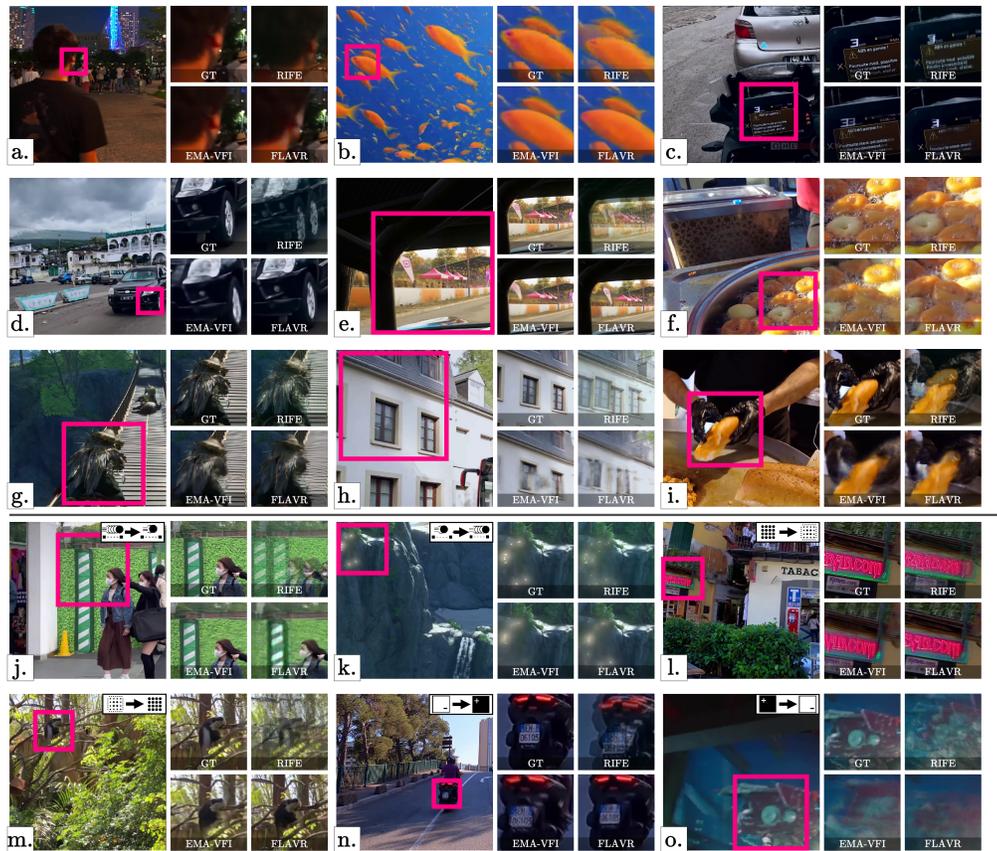
Figure 3: **Examples from the LAVIB benchmark and OOD test sets** (best viewed digitally). Zoomed regions on the right of each frame show interpolations with RIFE, EMA-VFI, and FLAVR. The top row shows results for videos from the benchmark test split. The bottom two rows are video frames from test splits from OOD challenges. The challenge is denoted at the top right of each ground truth frame. The ground truth is shown as a reference at the top left of the zoomed-in region grid.

Knowledge distillation. Transferring knowledge about the video structure can enable more efficient models. Several works [25, 35, 48] have distilled representations from teacher models trained on high-resolution videos. A natural extension of the proposed dataset would be its use for training high-resolution teacher models and evaluating low-resolution student models.

Salient frame sampling. The selection of informative frames has been another domain of interest for real-time video processing [66, 69, 70]. The standardized framerate of LAVIB can provide a robust benchmark for testing sampling approaches over different granularities.

Based on these adjacent video tasks, LAVIB can be imported and adapted as a general-purpose dataset and benchmark.

References

[1] Davide Abati, Haitam Ben Yahia, Markus Nagel, and Amirhossein Habibian. Resq: Residual quantization for video perception. In *ICCV*, pages 17119–17129, 2023. 9

[2] Simon Baker, Daniel Scharstein, James P Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *IJCV*, pages 1–31, 2011. 1, 2

[3] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *CVPR*, pages 3703–3712, 2019. 1

[4] AC Bovik, R Soundararajan, and Christos Bampis. On the robust performance of the st-rred video quality predictor, 2017. 2

[5] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *CVPR*, pages 5972–5981, 2022. 1

[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[7] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *AAAI*, pages 10663–10671, 2020. 2, 6

[8] Steffen Czolbe, Oswin Krause, Ingemar Cox, and Christian Igel. A loss function for generative neural networks based on watson's perceptual model. *NeurIPS*, pages 2051–2061, 2020. 2, 7

[9] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *TPAMI*, pages 2567–2581, 2020. 2, 7

[10] Tianyu Ding, Luming Liang, Zhihui Zhu, and Ilya Zharkov. Cdfi: Compression-driven network design for frame interpolation. In *CVPR*, pages 8001–8011, 2021. 1

[11] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. *NeurIPS*, pages 35946–35958, 2022. 5

[12] Qiqi Hou, Abhijay Ghildyal, and Feng Liu. A perceptual quality metric for video frame interpolation. In *ECCV*, pages 234–253, 2022. 2, 7

[13] Mengshun Hu, Kui Jiang, Zhixiang Nie, and Zheng Wang. You only align once: Bidirectional interaction for spatial-temporal video super-resolution. In *ACM-MM*, pages 847–855, 2022. 1

[14] Cong Huang, Jiahao Li, Bin Li, Dong Liu, and Yan Lu. Neural compression-based feature learning for video restoration. In *CVPR*, pages 5872–5881, 2022. 1

[15] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *ECCV*, pages 668–685, 2022. 2, 3, 5, 6

[16] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *ECCV*, pages 624–642, 2022. 1, 2, 6, 8, 14

[17] Takashi Isobe, Xu Jia, Xin Tao, Changlin Li, Ruihuang Li, Yongjie Shi, Jing Mu, Huchuan Lu, and Yu-Wing Tai. Look back and forth: Video super-resolution with explicit temporal difference modeling. In *CVPR*, pages 17411–17420, 2022. 1

[18] Xiaozhong Ji, Yun Cao, Ying Tai, Chengjie Wang, Jilin Li, and Feiyue Huang. Real-world super-resolution via kernel estimation and noise injection. In *CVPRw*, pages 466–467, 2020. 1

[19] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, pages 9000–9008, 2018. 1

[20] Meiguang Jin, Zhe Hu, and Paolo Favaro. Learning to extract flawless slow motion from blurry videos. In *CVPR*, pages 8112–8121, 2019. 1

[21] Xin Jin, Longhai Wu, Jie Chen, Youxin Chen, Jayoon Koo, and Cheul-hee Hahm. A unified pyramid recurrent network for video frame interpolation. In *CVPR*, pages 1578–1587, 2023. 1

[22] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, pages 3224–3232, 2018. 1

[23] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. Flavr: Flow-agnostic video representations for fast frame interpolation. In *WACV*, pages 2071–2082, 2023. 2, 6, 7, 8, 14

[24] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014. 2

[25] Hanul Kim, Mihir Jain, Jun-Tae Lee, Sungrack Yun, and Fatih Porikli. Efficient action recognition via dynamic knowledge propagation. In *ICCV*, pages 13719–13728, 2021. 10

[26] Taewoo Kim, Yujeong Chae, Hyun-Kurl Jang, and Kuk-Jin Yoon. Event-based video frame interpolation with cross-modal asymmetric bidirectional motion fields. In *CVPR*, pages 18032–18042, 2023. 1

[27] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomáš Vojíř, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *TPAMI*, pages 2137–2155, 2016. 2

[28] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *ICCV*, pages 8878–8887, 2019. 1

[29] Dingquan Li, Tingting Jiang, and Ming Jiang. Quality assessment of in-the-wild videos. In *ACM-MM*, pages 2351–2359, 2019. 2, 7

[30] Haopeng Li, Yuan Yuan, and Qi Wang. Video frame interpolation via residue refinement. In *ICASSP*, pages 2613–2617, 2020. 1

[31] Yinxiao Li, Pengchong Jin, Feng Yang, Ce Liu, Ming-Hsuan Yang, and Peyman Milanfar. Comisr: Compression-informed video super-resolution. In *ICCV*, pages 2543–2552, 2021. 1

[32] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. Vrt: A video restoration transformer. *T-IP*, pages 2171–2182, 2024. 1

[33] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc V Gool. Recurrent video restoration transformer with guided deformable attention. *NeurIPS*, pages 378–393, 2022. 1

[34] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *TPAMI*, pages 346–360, 2013. 2

[35] Chuofan Ma, Qiushan Guo, Yi Jiang, Ping Luo, Zehuan Yuan, and Xiaojuan Qi. Rethinking resolution in the context of efficient video recognition. *NeurIPS*, pages 37865–37877, 2022. 10

[36] Arthur B Markman and Dedre Gentner. Nonintentional similarity processing. *The new unconscious*, pages 107–137, 2005. 7

[37] Reece Mazade, Jianzhong Jin, Hamed Rahimi-Nasrabadi, Sohrab Najafian, Carmen Pons, and Jose-Manuel Alonso. Cortical mechanisms of visual brightness. *Cell reports*, pages 111438–111438, 2022. 4

[38] Christopher Montgomery and H Lars. Xiph. org video test media (derf's collection). *Online, https://media. xiph. org/video/derf*, 6, 1994. 1, 2, 3, 6, 7

[39] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPRw*, pages 1996–2005, 2019. 2

[40] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, pages 3883–3891, 2017. 1, 2

[41] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *CVPR*, pages 5437–5446, 2020. 1, 2, 3, 6, 7

[42] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, pages 261–270, 2017. 1

[43] Avinash Paliwal, Andrii Tsarov, and Nima Khademi Kalantari. Implicit view-time interpolation of stereo videos using multi-plane disparities and non-uniform coordinates. In *CVPR*, pages 888–898, 2023. 1

[44] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *ICCV*, pages 14539–14548, 2021. 1

[45] Eli Peli. Contrast in complex images. *JOSA A*, pages 2032–2040, 1990. 4

[46] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016. 1

[47] Charles Poynton. *Digital video and HD: Algorithms and Interfaces*. Elsevier, 2012. 4

[48] Didik Purwanto, Rizard Renanda Adhi Pramono, Yie-Tarng Chen, and Wen-Hsien Fang. Extreme low resolution action recognition with spatial-temporal multi-head self-attention and knowledge distillation. In *ICCVw*, pages 0–0, 2019. 10

[49] Hamed Rahimi-Nasrabadi, Veronica Moore-Stoll, Jia Tan, Stephen Dellostritto, JianZhong Jin, Mitchell W Dul, and Jose-Manuel Alonso. Luminance contrast shifts dominance balance between on and off pathways in human vision. *Journal of Neuroscience*, pages 993–1007, 2023. 4

[50] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *ECCV*, pages 250–266, 2022. 1

[51] Denys Rozumnyi, Martin R Oswald, Vittorio Ferrari, Jiri Matas, and Marc Pollefeys. Defmo: Deblurring and shape recovery of fast moving objects. In *CVPR*, pages 3456–3465, 2021. 2

[52] Wentao Shangguan, Yu Sun, Weijie Gan, and Ulugbek S Kamilov. Learning cross-video neural representations for high-quality frame interpolation. In *ECCV*, pages 511–528, 2022. 1

[53] Dev Yashpal Sheth, Sreyas Mohan, Joshua L Vincent, Ramon Manzorro, Peter A Crozier, Mitesh M Khapra, Eero P Simoncelli, and Carlos Fernandez-Granda. Unsupervised deep video denoising. In *ICCV*, pages 1759–1768, 2021. 1

[54] Shuwei Shi, Jinjin Gu, Liangbin Xie, Xintao Wang, Yujiu Yang, and Chao Dong. Rethinking alignment in video super-resolution transformers. *NeurIPS*, pages 36081–36093, 2022. 1

[55] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: extreme video frame interpolation. In *ICCV*, pages 14489–14498, 2021. 1, 2, 3, 6, 7

[56] Li Siyao, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris Metaxas, Chen Change Loy, and Ziwei Liu. Deep animation video interpolation in the wild. In *CVPR*, pages 6587–6595, 2021. 1, 2

[57] Ronald D Snee. Validation of regression models: methods and examples. *Technometrics*, pages 415–428, 1977. 6

[58] Mingyang Song, Yang Zhang, and Tunç O Aydın. Tempformer: Temporally consistent transformer for video denoising. In *ECCV*, pages 481–496, 2022. 1

[59] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 1, 3, 6

[60] Alexandros Stergiou and Ronald Poppe. Adapool: Exponential adaptive pooling for information-retaining downsampling. *T-IP*, pages 251–266, 2022. 1, 2, 3

[61] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *CVPR*, pages 1279–1288, 2017. 1

[62] Ximeng Sun, Rameswar Panda, Chun-Fu Richard Chen, Aude Oliva, Rogerio Feris, and Kate Saenko. Dynamic network quantization for efficient video inference. In *ICCV*, pages 7375–7385, 2021. 9

[63] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *ICCV*, pages 4472–4480, 2017. 1

[64] Matias Tassano, Julie Delon, and Thomas Veit. Dvdnet: A fast network for deep video denoising. In *ICIP*, pages 1805–1809, 2019. 1

[65] Longguang Wang, Yulan Guo, Li Liu, Zaiping Lin, Xinpu Deng, and Wei An. Deep video super-resolution using hr optical flow estimation. *T-IP*, pages 4323–4336, 2020. 1

[66] Yulin Wang, Yang Yue, Xinhong Xu, Ali Hassani, Victor Kulikov, Nikita Orlov, Shiji Song, Humphrey Shi, and Gao Huang. Adafocusv3: On unified spatial-temporal dynamic video recognition. In *ECCV*, pages 226–243, 2022. 10

[67] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *T-IP*, pages 600–612, 2004. 7

[68] Zhongyuan Wang, Peng Yi, Kui Jiang, Junjun Jiang, Zhen Han, Tao Lu, and Jiayi Ma. Multi-memory convolutional neural network for video super-resolution. *T-IP*, pages 2530–2544, 2018. 2

[69] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. Adaframe: Adaptive frame selection for fast video recognition. In *CVPR*, pages 1278–1287, 2019. 10

[70] Boyang Xia, Wenhao Wu, Haoran Wang, Rui Su, Dongliang He, Haosen Yang, Xiaoran Fan, and Wanli Ouyang. Nsnet: Non-saliency suppression sampler for efficient video recognition. In *ECCV*, pages 705–723, 2022. 10

[71] Dan Xu, Andrea Vedaldi, and Joao F Henriques. Moving slam: Fully unsupervised deep learning in non-rigid scenes. In *IROS*, pages 4611–4617, 2021. 1

[72] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *IJCV*, pages 1106–1125, 2019. 1, 2, 3, 6, 7

[73] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, and Jiayi Ma. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *ICCV*, pages 3106–3115, 2019. 2

[74] Jun-Sang Yoo, Hongjae Lee, and Seung-Won Jung. Video object segmentation-aware video frame interpolation. In *ICCV*, pages 12322–12333, 2023. 1

[75] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *CVPR*, pages 5682–5692, 2023. 1, 2, 6, 8, 14

[76] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 2, 5, 6, 7

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section TODO.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The paper introduces a dataset, video selection pipeline, and challenges. Both the main paper and appendix, thoroughly describe the annotation pipeline, metrics used, results obtained, and comparison made to other datasets.
   (b) Did you describe the limitations of your work? [Yes] Limitations at each step of the data collection are discussed in §4.1 and §4.2.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] A section in the appendix is devoted to potential data biases and resulting models.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Train/val/test splits for the benchmark and all OOD challenges and settings are publicly available: https://alexandrosstergiou.github.io/datasets/LAVIB. The code is published on GitHub.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Model settings and training details and splits are discussed in Section 5. Further details on the training settings are discussed in §A3 of the appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Experiments are too computationally costly to do multiple runs. Due to the size of the dataset, only small variations in performance are expected with different seeding. Reported results form baselines for which stronger future methods are expected to outperform.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Compute resources are detailed at §A3 of the appendix.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] LAVIB is sourced from YouTube videos. Baselines use models and the codebases of RIFE [16], EMA-VFI [75], and FLAVR [23]. The adjusted codebases for running LAVIB are available at https://github.com/alexandrosstergiou/LAVIB.
   (b) Did you mention the license of the assets? [No] All spatiotemporal cropped video segments used are from publicly available data, following the Terms of Service users agreed to when uploading to YouTube. Codebases for the models used include open-source licenses.

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Videos are publicly available. §A6 contains more details.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Some videos include recordings of people, albeit occasionally, they can sometimes include their faces briefly. This is discussed further in §A6.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]