Generalizable Video Quality Assessment via Weak-to-Strong Learning

Supplementary Material

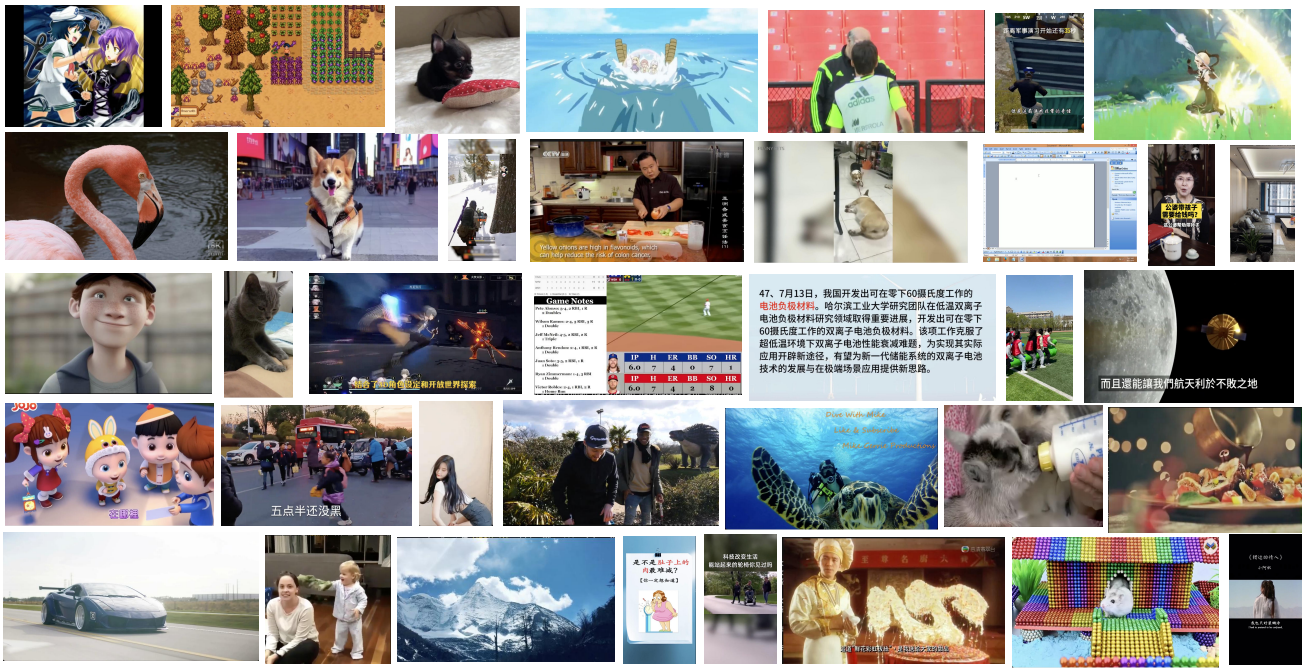

Figure 1. Examples of videos from different categories in our large dataset.

A. More Details of Our D_{w2s} Database

A.1. Analysis of the Collected Videos

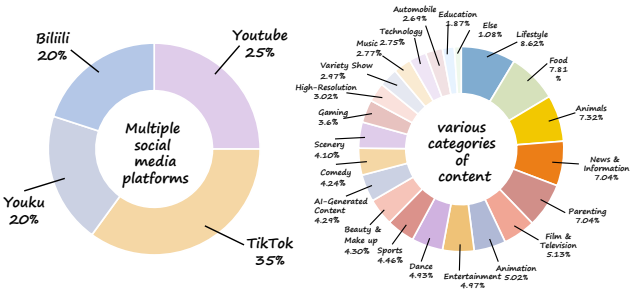

Figure 2. Our dataset is collected from multiple popular social media platforms and encompasses a wide range of content categories.

As shown in Fig. 2, our dataset is collected from multiple popular social media platforms with relatively uniform sampling, comprising 20% from Bilibili, 20% from Youku, 25% from YouTube, and 35% from TikTok. **All videos are obtained through a filtering pipeline that ensures only publicly available content with permissive licenses is included.** Notably, our dataset covers a diverse range of content categories, exceeding twenty in total. In addition to common categories such as lifestyle, food, and animals, it also includes specialized categories such as gaming, AI-generated content, and high-resolution content. To illustrate the diversity of our dataset, we present a variety of video samples in Fig. 1, showcasing the broad range of content available in our large-scale video quality assessment (VQA) dataset. Unlike existing datasets, which often focus on specific formats, our dataset encompasses a wider variety of formats, including both landscape and portrait orientations, as well as various resolutions. This diversity enhances the comprehensiveness of our dataset, making it more suitable for evaluating video quality across a wide range of scenarios. A detailed breakdown of our database, including pair types and the corresponding number of videos, is provided in Table 1.

Table 1. Statistics of raw videos and video pairs in the D_{w2s} dataset.

Category	Subtype	Videos			Video Pairs		
		$D^{(1)}_{\text{w2s}}$	$D^{(2)}_{\text{w2s}}$	$D^{(3)}_{\text{w2s}}$	$D^{(1)}_{\text{w2s}}$	$D^{(2)}_{\text{w2s}}$	$D^{(3)}_{\text{w2s}}$
Ensembling homogeneous teachers	-	200k	100k	50k	250k	85k	85k
Integrating heterogeneous teachers	Spatial	50k	2k	2k	160k	5k	5k
	Temporal	20k	1k	1k	40k	5k	5k
	Compression	10k	1k	1k	50k	5k	5k
Total		280k	384k	438k	500k	600k	700k

A.2. Analysis of Low-level Metrics

To ensure that our constructed dataset exhibits sufficient diversity across various low-level visual characteristics, we adopt a metric-guided sampling strategy. Specifically, we compute nine commonly used low-level metrics—blockiness [16], blur [13], contrast [15], noise, flickering [14], colourfulness [5], luminance, spatial information (SI) [7], and temporal information (TI) [7]. These metrics are employed to guide data sampling by covering a wide range of values in each dimension, thereby promoting diversity in visual content and distortion patterns. The distribution of nine metrics on our dataset before and after sampling is shown in Figure 3. Each metric is computed as follows:

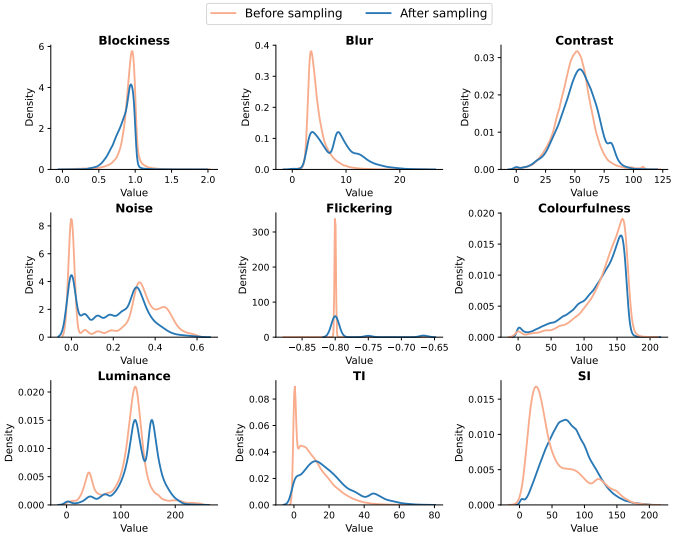

Figure 3. Distribution of nine metrics on our dataset before and after sampling.

Blockiness [16] is quantified by analyzing the luminance differences between pixels within and across encoding blocks. Specifically, we compute the absolute luminance differences between adjacent pixel pairs within the same encoding block (internal pixel pairs) and those spanning adjacent blocks (external pixel pairs). The blockiness metric is then determined as the ratio of the total sum of internal pixel difference values to the total sum of external pixel difference values across the entire video frame:

$$B = \frac{\sum_{(x,y)\in\mathcal{I}} |I(x,y) - I(x+1,y)|}{\sum_{(x,y)\in\mathcal{E}} |I(x,y) - I(x+1,y)|}, \tag{1}$$

where $I(x,y)$ represents the luminance value at pixel location (x,y), $\mathcal{I}$ denotes the set of internal pixel pairs, and $\mathcal{E}$ represents the set of external pixel pairs. A higher blockiness value indicates stronger blocking artifacts, which typically result from aggressive video compression.

Blur is measured using the Cumulative Probability of Blur Detection (CPBD) [13], which evaluates perceptual sharpness based on edge width distribution. A higher CPBD value indicates a sharper image. Given an edge pixel e_i, its width $w(e_i)$ is compared with the Just Noticeale Blur (JNB) threshold, determining the blur detection probability $w_{JNB}(e_i)$. The final CPBD score is computed as:

$$\text{CPBD} = P(P_{\text{BLUR}} \leq P_{\text{JNB}}) = \sum_{P_{\text{BLUR}}=0}^{P_{\text{JNB}}} P(P_{\text{BLUR}}). \tag{2}$$

Contrast is a measure of the dispersion of pixel intensity values within the video frame and can be quantified using the standard deviation of grayscale intensities [15]. Specifically, for a grayscale image $I(x,y)$, the mean intensity μ is first computed as:

$$\mu = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} I(x,y), \tag{3}$$

where M and N denote the width and height of the image, respectively, and $I(x,y)$ represents the intensity at pixel (x,y). The contrast value σ is then obtained by calculating the standard deviation of intensity values:

$$\sigma = \sqrt{\frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} (I(x,y) - \mu)^2}. \tag{4}$$

The standard deviation σ represents the contrast of the video frame, where a higher σ value indicates a greater dispersion of intensity values and thus a higher contrast.

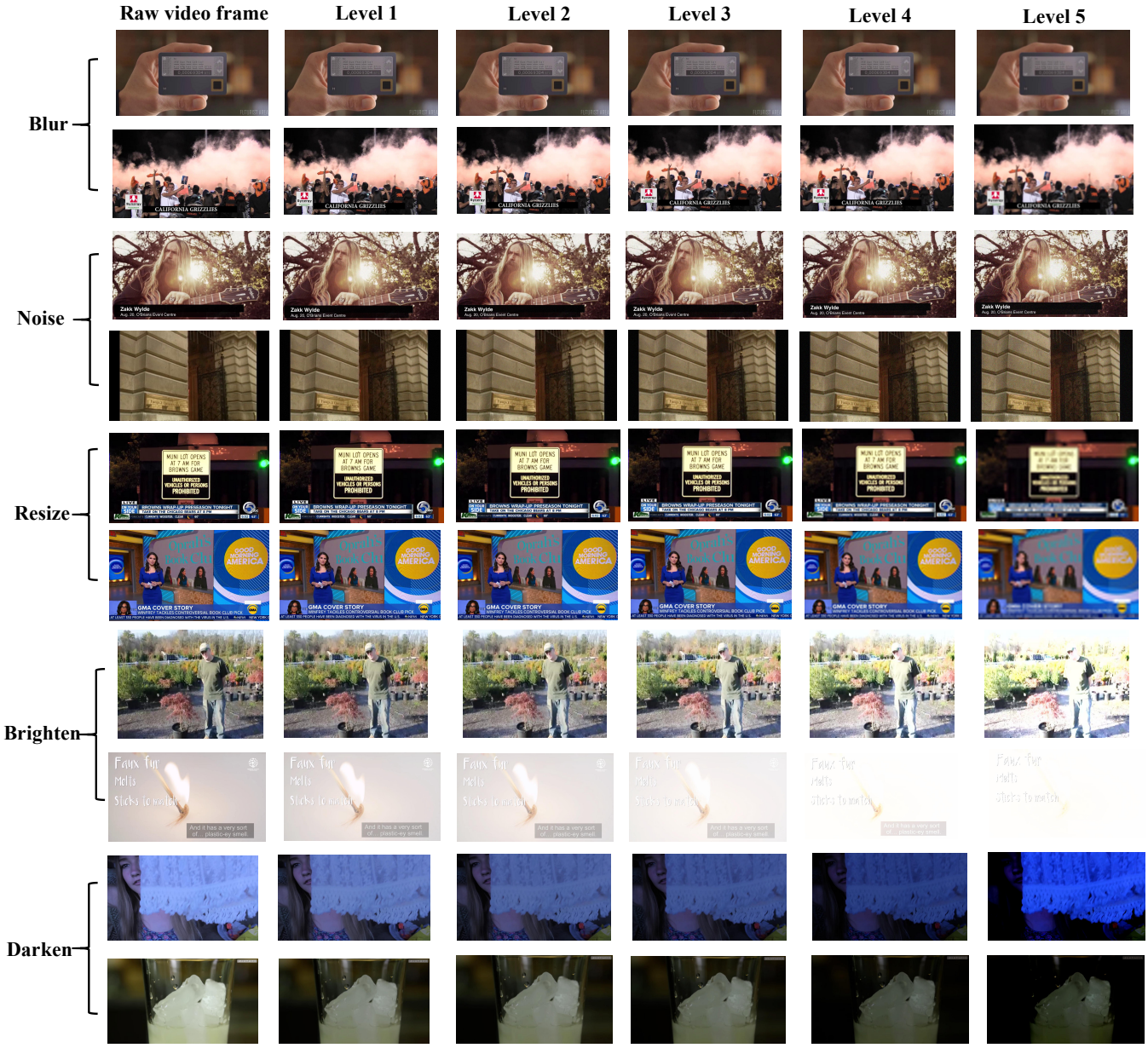

Figure 4. Illustration of different levels of spatial distortion video frames in our dataset.

Noise refers to random intensity fluctuations that do not originate from the underlying scene content. It is measured by estimating the high-frequency residual that remains after removing the structural component of the frame. Given a frame $I(x, y)$, a smoothed version $\hat{I}(x, y)$ is first obtained using a low-pass filter. The noise residual is computed as:

$$R(x, y) = I(x, y) - \hat{I}(x, y), \tag{5}$$

and the noise metric is defined as the normalized standard deviation of this residual:

$$Noise = \frac{1}{\sigma_{\max}} \sqrt{\frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} R(x, y)^2}, \tag{6}$$

where $\sigma_{\max}$ is a normalization constant.

Flickering occurs when an encoder skips macroblocks to conserve bitrate, especially in low-texture, slow-motion regions [14]. It is quantified by counting macroblock transitions from an "unupdated" to an "updated" state, with a

threshold T_f ensuring only significant changes are considered. The flickering metric is computed as:

$$F = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} \mathbb{I}\left(|I_t(x,y) - I_{t-1}(x,y)| > T_f\right), \tag{7}$$

where $I_t(x,y)$ is the luminance at pixel (x,y) in frame t, and $\mathbb{I}(\cdot)$ is an indicator function. A higher F indicates stronger flickering artifacts.

Colourfulness quantifies color distribution differences across RGB channels, following [5]. Given a frame with RGB channels R, G, B, we compute:

$$r_g = R - G, \quad y_b = \frac{1}{2}(R+G) - B. \tag{8}$$

The Colourfulness metric is then:

$$C = \sqrt{\sigma_{r_g}^2 + \sigma_{y_b}^2} + 0.3 \times \sqrt{\mu_{r_g}^2 + \mu_{y_b}^2}, \tag{9}$$

where σ and μ denote the standard deviations and means of r_g and y_b, respectively.

Luminance is measured as the combined intensity of the three RGB channels, defined as:

$$L = R + G + B. \tag{10}$$

SI measures spatial complexity using the Sobel filter. The standard deviation of the Sobel-filtered frame over all pixels is computed, and the maximum value over time represents the SI:

$$SI = \max_{time} \left\{ \text{std}_{space} \left[\text{Sobel}(F_n) \right] \right\}. \tag{11}$$

TI measures motion intensity by calculating the difference between consecutive frames. The temporal difference at pixel (i,j) is:

$$M_n(i,j) = F_n(i,j) - F_{n-1}(i,j). \tag{12}$$

The TI value is the maximum standard deviation of $M_n(i,j)$ over time and space:

$$TI = \max_{time} \left\{ \text{std}_{space} [M_n(i,j)] \right\}. \tag{13}$$

To optimize computational efficiency, all metrics are extracted at a sampling rate of one frame per second.

A.3. More Details on Synthetic Distortion Data

A.3.1. Spatial Distortions

We introduce five common spatial distortions: resizing, Gaussian blur, Gaussian noise, darkening, and brightening. Each distortion is applied at five different levels to simulate varying degrees of degradation, ranging from mild to severe. Fig. 4 illustrates examples of these distortions, where the quality of video frames progressively deteriorates as the distortion level increases. Below, we provide details on how these spatial distortions are generated, where I represents the original frame, and I' denotes the distorted frame.

Resizing: The frame is first downsampled by a scaling factor s and then upsampled back to its original size. This process reduces spatial details and introduces pixelation artifacts, simulating resolution loss. The transformation is defined as:

$$I' = \text{Upsample}(\text{Downsample}(I, s), s), \tag{14}$$

where s takes values from the set $\{2, 3, 4, 8, 16\}$.

Gaussian Blur: The frame is convolved with a Gaussian kernel, where the standard deviation σ_{blur} controls the extent of the blur. A larger σ_{blur} results in a wider spread of the Gaussian function, leading to a stronger blurring effect by averaging pixel intensities over a larger neighborhood. The blurring process is defined as:

$$I' = I * G(\sigma_{blur}), \tag{15}$$

where $G(\sigma_{blur})$ is a Gaussian kernel with standard deviation σ_{blur} which takes values from the set $\{0.1, 0.5, 1, 2, 5\}$, and $*$ denotes the convolution operation.

Gaussian noise: Gaussian noise is introduced by adding random variations to each pixel, following a normal distribution with mean μ and standard deviation σ_{noise}. The noise level is controlled by adjusting σ_{noise}, where higher values result in more pronounced noise artifacts. The process is defined as:

$$I' = I + N(\mu, \sigma_{noise}^2), \tag{16}$$

where $N(\mu, \sigma_{noise}^2)$ represents Gaussian noise with mean μ and variance σ_{noise}^2, added independently to each pixel. σ takes values from the set $\{0.001, 0.002, 0.003, 0.005, 0.01\}$.

Darkening: Darkening is applied by reducing the luminance component in the color space. The effect is controlled by a parameter p, which determines the degree of brightness reduction. The luminance channel L is adjusted using an interpolation function $f(L, p)$ as follows:

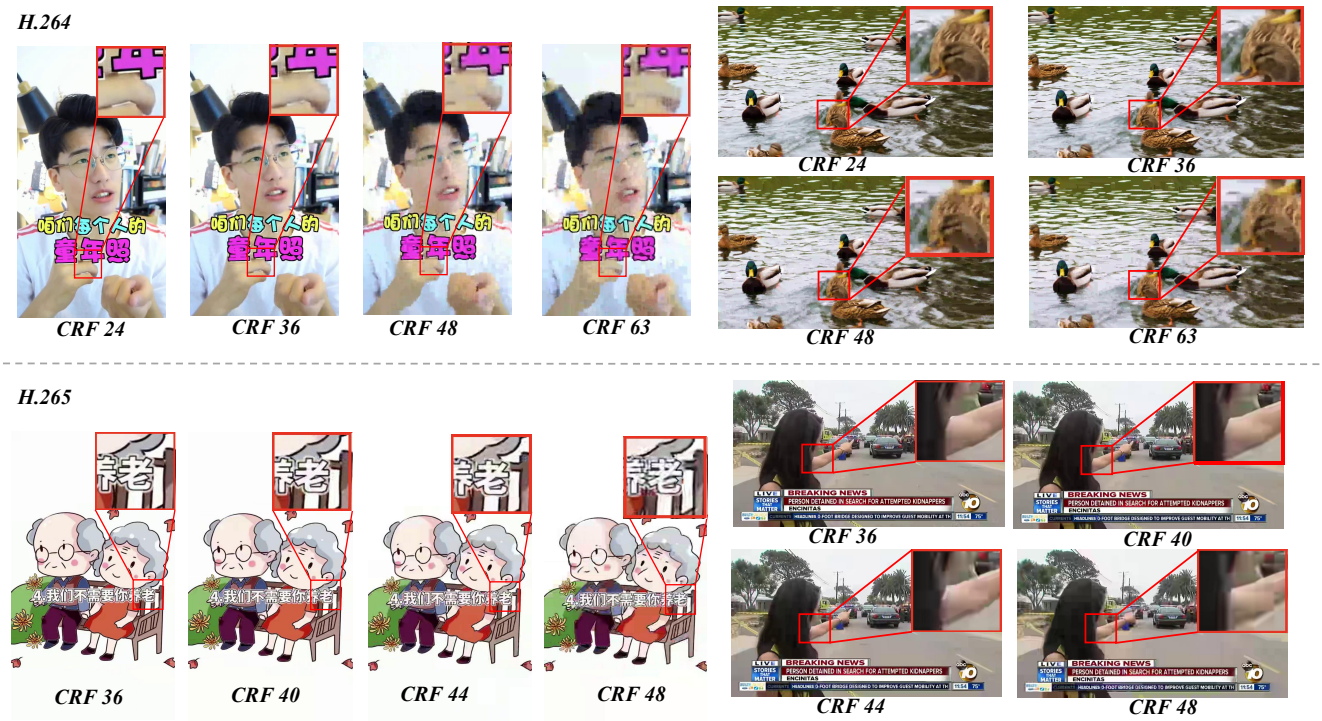

Figure 5. Illustration of different levels of streaming distortion video frames in our dataset.

$$L' = f(L, p). \tag{17}$$

The parameter p is selected from a predefined set of values $\{0.05, 0.1, 0.2, 0.4, 0.8\}$, with larger values leading to stronger darkening effects.

Brightening: In contrast, brightening is achieved by enhancing the luminance component in the color space. The luminance channel L is modified using a nonlinear transformation function $g(L, p)$:

$$L' = g(L, p), \tag{18}$$

The parameter p is selected from $\{0.1, 0.2, 0.4, 0.7, 1.1\}$, with larger values producing a stronger brightening effects.

A.3.2. Temporal Distortions

We introduce two types of temporal distortions: jitter and stuttering, each distortion maintain three different levels.

Jitter: Jitter introduces random shifts and random cropping followed by resizing of video frames. The amount of shift is determined by the jitter level, which controls the extent of spatial displacement.

For each frame, random horizontal and vertical shifts are applied using an affine transformation matrix, which shifts the frame along the x- and y-axes. Additionally, each frame is cropped by a small amount from the edges and resized back to its original dimensions, simulating pixelation effects or lower-quality views. The transformation matrix is described as follows:

$$M = \begin{bmatrix} 1 & 0 & \text{random\_shift\_x} \\ 0 & 1 & \text{random\_shift\_y} \end{bmatrix} \tag{19}$$

where random_shift_x and random_shift_y are random values determined by the jitter level.

Stuttering: Stuttering is introduced by randomly dropping frames at a controlled rate. The drop rate p_d is determined by the distortion level, where higher levels correspond to increased frame loss. For each frame I_t, a random probability is drawn and compared with p_d. If the frame is dropped, it is replaced by the previous frame I_{t-1}, simulating temporal freezing in the video. The process can be formulated as:

$$I'_t = \begin{cases} I_{t-1}, & \text{if } r < p_d, \\ I_t, & \text{otherwise} \end{cases} \tag{20}$$

where $r \sim U(0, 1)$ is a random variable drawn from a uniform distribution.

Table 2. An overview of our testing datasets.

Dataset	Year	# of Videos	# of Scenes	Resolution	Duration	Frame Rate	Distortion Type
KoNViD-1k [6]	2017	1,200	1,200	540p	8	24, 25, 30	In-the-wild
LIVE-VQC [18]	2018	585	585	240p–1080p	10	30	In-the-wild
YouTube-UGC [22]	2019	1,380	1,380	360p–4K	20	30	In-the-wild
LSVQ [26]	2021	38,811	38,811	99p–4K	5–12	< 60	In-the-wild
Waterloo-IVC-4K [9]	2019	1200	20	540p, 1080p, 4k	9-10	24, 25, 30	H.264 compression
LIVE-YT-HFR [12]	2021	480	16	1080p	6-10	24, 30, 60, 82, 98, 120	Frame rate, VP9 compression
LIVE-YT-Gaming [27]	2022	600	600	360p–1080p	8–9	30, 60	PGC, UGC
CGVDS [17]	2023	360	15	480p, 720p, 1080p	30	20, 30, 60	H.264 compression
KVQ [11]	2024	4200	600	-	3-8	-	UGC

A.3.3. Streaming Distortions

As illustrated in Fig. 5, we select the two most common compression standards, H.264 and H.265, to simulate video quality degradation for the compression distortion. These distortions are applied using the `ffmpeg` tool, a widely used multimedia framework, to encode the videos with different compression settings. Specifically, we chose four fixed constant rate factor (CRF) values for each compression standard to control the level of distortion.

For H.264 compression, we selected the `fast` encoding mode, which provides a good balance between encoding speed and compression efficiency, making it suitable for real-time applications. To cover a wide range of compression levels, we applied H.264 compression using CRF values of 24, 36, 48, and 63, ensuring the simulation of various quality degradation scenarios.

In contrast, for H.265 compression, we selected the `very slow` encoding mode, which prioritizes compression efficiency over speed, leading to higher quality video at the cost of longer encoding times. To achieve fine-grained quality simulation, we applied H.265 compression with a narrower CRF range of 36, 40, 44, and 48, allowing for precise control over compression artifacts.

These encoding settings help to simulate typical real-world compression scenarios, where different modes and CRF values are chosen based on the trade-off between video quality and encoding performance.

A.4. More Details on Testing Datasets

Table 2 provides an overview of our testing datasets, which encompass diverse content types, resolutions, durations, frame rates, and distortion types. The first four datasets consist of in-the-wild videos containing various authentic distortions, while the remaining datasets focus on specific content types and distortion factors. For example, LIVE-YT-Gaming is dedicated to gaming content, LIVE-YT-HFR targets frame rate distortions, and Waterloo-IVC-4K covers different types of compression artifacts. By evaluating our model across these nine datasets, we demonstrate its robustness and effectiveness in both in-domain and out-of-distribution (OOD) quality assessment scenarios.

B. More Details of Quality Annotation

B.1. Weak Models for Pseudo-labeling

Table 3. Comparison of model parameters and architecture.

Model	Parameters (M)	Architecture
MinimalisticVQA(VII)	86.93	Swin-B
MinimalisticVQA (IX)	121.59	Swin-B + SlowFast
FAST-VQA	29.97	Swin-Tiny
DOVER	58.06	Swin-Tiny + Conv-Tiny
Q-Align	8204.56	mPLUG-Owl2
Our strong model	8075.24	LLaVA-OneVision-Chat + SlowFast

We choose five SOTA VQA models: MinimalisticVQA (VII) [19], MinimalisticVQA (IX) [19], FAST-VQA [23], DOVER [24], and Q-Align [25] as weak teachers to formulate our pseudo quality annotation. The detail introduction of the five models is as follows:

MinimalisticVQA (VII) employs Swin Transformer-B [10], pre-trained on ImageNet-1K [3], as the spatial quality analyzer to extract quality-aware spatial features from key frames, ensuring robust spatial quality assessment.

MinimalisticVQA (IX) builds upon MinimalisticVQA (VII) by incorporating a temporal quality analyzer to account for motion distortions. The temporal quality analyzer, implemented using the SlowFast [4] network pre-trained on the Kinetics-400 [2] dataset, extracts motion-related features from video chunks, enhancing the model's ability to assess temporal quality variations.

FAST-VQA introduces Grid Mini-patch Sampling (GMS) strategy, which preserves local quality by sampling patches at raw resolution and maintains global quality through uniformly sampled mini-patches. These mini-patches are spliced and temporally aligned into fragments. To process these fragments, the Fragment Attention Network (FANet) is designed to effectively extract video quality features. Combining GMS and FANet, FAST-VQA achieves efficient end-to-end video quality assessment with effective feature representation learning.

DOVER builds upon FAST-VQA as its technical branch to capture low-level distortions, while introducing an additional aesthetic branch to assess high-level semantic composition, which relates to user preferences and content recommendation. By disentangling these two perspectives, DOVER establishes a more human-aligned and interpretable framework for video quality assessment.

Q-Align presents a novel training strategy for large multimodal model (LMM) in VQA by replacing direct numerical score predictions with discrete, text-defined rating levels (e.g., "excellent", "good", "fair", "poor", "bad") as learning targets. During inference, Q-Align extracts the log probabilities of each rating level, applies softmax normalization to obtain a probability distribution, and computes a weighted average to derive the final predicted quality score.

We choose this set of weak teachers for three reasons:

- They represent widely adopted and highly competitive VQA paradigms, covering spatial–temporal modeling, efficient convolutional designs, transformer-based architectures, and multimodal alignment.
- Their computational overhead remains relatively low, making large-scale pseudo-label generation feasible for millions of videos.
- Using multiple weak models allows us to obtain more comprehensive and less biased pseudo-supervision than relying on a single teacher; therefore, we select a set of five weak models.

It is worth noting that our framework does not depend on these specific five model, and other VQA models can be readily substituted. Our goal is to provide a general strategy for constructing strong homogeneous pseudo-supervision from diverse weak sources, rather than asserting this particular set as canonical.

We use a four-parameter logistic function to map the predicted scores from different weak models onto a common scale for subsequent evaluation. For each model, we first collect its raw predictions $\{y_i\}$ on the *LSVQ test subset* and fit a four-parameter logistic mapping that relates these predictions to the corresponding ground-truth quality scores $\{g_i\}$. The calibration function is given by

$$f(s) = \beta_2 + \frac{\beta_1 - \beta_2}{1 + \exp\left(-\frac{s-\beta_3}{|\beta_4|}\right)},$$

where $(\beta_1, \beta_2, \beta_3, \beta_4)$ are obtained by minimizing the least-squares error between $f(y_i)$ and g_i over the test set. Once fitted, this monotonic transformation is applied to all prediction scores produced by the same model:

$$\tilde{y}_i = f(y_i),$$

thereby aligning the model's entire score range with the empirical label distribution of *LSVQ test subset*. This procedure ensures that prediction scales of different models become consistent, enabling fair and meaningful cross-model evaluation.

B.2. Prompts for Model Training

We construct the label prompts for our large-scale dataset using a fixed template. For the single-video input:

```
Question: "You will now receive
a video: <image>. Please watch
the video carefully and answer the
following question: What is your
overall rating of the quality of
this video?"
Answer: "[quality score]"
```

For the dual-video input:

```
Question: "You will now receive
two videos. The first video:
<image>. The second video:
<image>. Please watch both videos
carefully and answer the following
question: Compared to the first
video, how would you rate the
quality of the second video?"
Answer: "The quality of the
second video is [level] compared
to the first video."
```

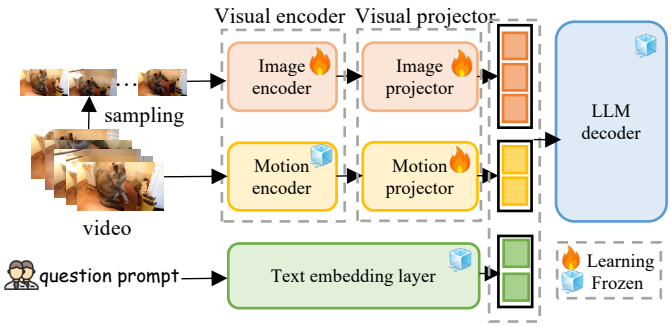

Figure 6. The overall structure of our model.

C. More Details of Our Strong student Model

C.1. Model Structure

As illustrated in Fig. 6, our model comprises three components: a visual feature extractor, a text tokenizer, and an LLM decoder.

Visual Feature Extractor. The visual feature extractor adopts a dual-branch design: a spatial branch with image encoder $\mathcal{F}_I$ (*i.e.,* SigLIP) processes key frames, while a

temporal branch with pre-trained motion encoder $\mathcal{F}_M$ (*i.e.,* SlowFast) analyzes frame sequences. Both branches employ dedicated projection layers $\mathcal{P}_\mathcal{I}$ and $\mathcal{P}_\mathcal{F}$ (*i.e.,* two-layer MLPs) to map spatial and temporal features into visual tokens aligned with language space. Specifically, given an input video $\boldsymbol{x} = \{\boldsymbol{x}_i\}_{i=0}^{N-1}$ containing N frames at frame rate r, we first partition it into $N_c = \lfloor N/r \rfloor$ continuous chunks $\{\boldsymbol{c}_k\}_{k=0}^{N_c-1}$, where each chunk $\boldsymbol{c}_k = \{x_j\}_{j=k*r}^{(k+1)*r}$ spans r frames. Spatial features $\boldsymbol{f}_k^s$ are extracted from the first frame $\boldsymbol{x}_{kr}$ of each chunk, while temporal features $\boldsymbol{f}_k^t$ are computed over all frames in c_k. The feature extraction process is formally expressed as:

$$\begin{aligned} &\boldsymbol{f}_k^s = \mathcal{P}_I(\mathcal{F}_I(\boldsymbol{x}_{kr})), \quad \boldsymbol{f}_k^t = \mathcal{P}_M(\mathcal{F}_M(\boldsymbol{c}_k)), \\ &\boldsymbol{f}^v = \text{Concat}\left([\boldsymbol{f}_k^s, \boldsymbol{f}_k^t]_{k=0}^{N_c-1}\right), \end{aligned} \tag{21}$$

where $\boldsymbol{f}^v$ is the extracted visual features of $\boldsymbol{x}$. Given a video pair $(\boldsymbol{x}^A, \boldsymbol{x}^B)$, we can derive the visual features $(\boldsymbol{f}_A^v, \boldsymbol{f}_B^v)$.

Feature Fusion via the LLM. Given an input prompt $\boldsymbol{p}$, we first encode it into text tokens $\boldsymbol{f}^p = \mathcal{T}(\boldsymbol{p})$ using tokenizer $\mathcal{T}$. The visual features of a video pair $(\boldsymbol{f}_A^v, \boldsymbol{f}_B^v)$ are then concatenated with $\boldsymbol{f}^t$ and fed to a pretrained LLM decoder (*i.e.,* Qwen-2) for multimodal fusion to derive the output response for quality ranking:

$$\boldsymbol{r} = \mathcal{L}(\boldsymbol{f}_A^v, \boldsymbol{f}_B^v, \boldsymbol{f}^p), \tag{22}$$

where $\boldsymbol{r}$ is expected to belong to {"superior", "better", "similar", "worse", "inferior"}.

C.2. Training Details

C.2.1. Training Setup

The model is trained using the DeepSpeed framework with mixed-precision floating-point operations to optimize memory and computational efficiency. The training is conducted for one epoch with a batch size of 1 per device and a gradient accumulation step of 1. The optimizer follows AdamW with a initial learning rate of 1×10^{-4}, a cosine learning rate schedule, and a warm-up ratio of 0.03.

We employ a joint training strategy for images and videos. For the image encoder, videos are sampled at a rate of one frame per second, with each sampled frame resized to a resolution of 384×384, while images are directly resized to the same resolution. For the motion encoder, videos are fully encoded across all frames to capture temporal dynamics, whereas images, which lack temporal information, are assigned an all-zero tensor as their temporal representation.

C.2.2. Auxiliary Confidence Loss

As mentioned in the main paper (Section 4.3), we introduce an auxiliary confidence loss to encourage the model to maintain high-confidence predictions, especially in the presence of noisy weak supervision. The final training objective is a dynamically weighted combination of the cross-entropy loss $\mathcal{L}_{\text{CE}}$ and the confidence loss $\mathcal{L}_{\text{conf}}$:

$$\mathcal{L} = (1-\lambda) \cdot \mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{L}_{\text{conf}}, \tag{23}$$

where λ is an adaptive weighting factor that balances between trusting the weak labels and relying on the model's own confidence. The confidence loss is defined as the average entropy over the predicted token probability distributions:

$$\mathcal{L}_{\text{conf}} = \frac{1}{N}\sum_{i=1}^{N} H(p_\theta(x_i)) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c} p_\theta(c|x_i)\log p_\theta(c|x_i), \tag{24}$$

where $p_\theta(c|x_i)$ denotes the predicted probability of vocabulary token c given input x_i. By minimizing the entropy of the predicted distribution, we encourage the model to produce more confident next-token predictions.

To dynamically adjust λ during training, we introduce a temperature-based confidence estimation mechanism. Specifically, we define:

$$\lambda = \alpha \cdot \min\left(1.0, \frac{t}{T_{\text{warmup}}}\right), \tag{25}$$

where t denotes the current training step ratio (normalized to $[0, 1]$), and T_{warmup} is the warm-up period, which we set to 10% of the total training steps. This warm-up phase ensures that the strong model gradually learns to rely on its own confidence, while initially being guided by the weak labels. The factor α is computed as the ratio between the temperature-scaled exponentials of the two losses:

$$\alpha = \frac{\exp(\mathcal{L}_{\text{conf}}/T)}{\exp(\mathcal{L}_{\text{conf}}/T) + \exp(\mathcal{L}_{\text{CE}}/T)}. \tag{26}$$

Here, T is a temperature parameter that controls the sharpness of the weighting between the two loss components. We linearly decrease T from 0.5 to 0.1 during the warm-up period to gradually increase the sensitivity of α to differences in the two loss values.

C.3. Inferring Details

C.3.1. Probability Modeling

Though we employ video pairs to train our model by enabling it to determine whether the second video is better than the first, our goal during inference is to obtain an absolute quality score for a single video. To achieve this, we propose a method that converts the probability of a test video being better or worse than anchor videos into a final quality score.

Table 4. Detailed performance comparison of weak teachers, students trained with weak teacher labels at two data scales (27k vs. 200k), and students trained with LSVQ ground-truth labels. Best performance in each category is indicated in **bold**.

In-domain Datasets	**$LSVQ_{test}$**		**$LSVQ_{1080p}$**		**KoNViD-1k**		**LIVE-VQC**		**YouTube-UGC**		**Overall**	
# of videos	7,182		3,573		1,200		585		1,020		-	
Methods	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**
Weak Teachers												
MinimalisticVQA(VII)	0.861	0.859	0.740	0.784	0.843	0.841	0.757	0.813	0.775	0.779	0.817	0.830
MinimalisticVQA(IX)	0.885	0.882	**0.792**	**0.828**	0.862	0.859	0.775	0.821	0.826	0.821	**0.849**	0.859
FAST-VQA	0.880	0.880	0.781	0.813	0.859	0.854	**0.826**	**0.845**	0.730	0.747	0.838	0.849
DOVER	0.878	0.866	0.782	0.813	0.874	0.869	0.817	0.840	0.771	0.781	0.842	0.845
Q-Align	**0.886**	**0.884**	0.761	0.822	**0.876**	**0.878**	0.783	0.819	**0.834**	**0.846**	0.844	**0.861**
Students Supervised by Weak Teacher Labels (27k)												
MinimalisticVQA(VII)	0.850	0.848	0.756	0.798	0.850	0.847	0.763	0.809	0.807	0.818	0.818	0.831
MinimalisticVQA(IX)	0.874	0.871	**0.793**	0.825	0.864	0.863	0.790	0.821	**0.835**	0.836	0.845	0.853
FAST-VQA	0.868	0.865	0.779	0.814	0.848	0.850	**0.793**	**0.829**	0.817	0.824	0.836	0.846
DOVER	0.873	0.866	0.779	0.815	0.863	**0.876**	0.778	0.819	0.824	0.831	0.840	0.849
Q-Align	**0.876**	**0.873**	0.792	**0.828**	**0.869**	0.869	0.786	0.828	0.827	**0.843**	**0.846**	**0.857**
Students Supervised by Weak Teacher Labels (200k)												
MinimalisticVQA(VII)-labeled	0.855	0.852	0.762	0.795	0.859	0.857	0.771	0.813	0.808	0.821	0.824	0.833
MinimalisticVQA(IX)-labeled	**0.879**	**0.878**	**0.794**	**0.826**	0.869	0.871	0.786	0.822	**0.843**	0.846	**0.849**	**0.859**
FAST-VQA-labeled	0.873	0.868	0.785	0.819	0.849	0.855	**0.798**	**0.833**	0.825	0.834	0.842	0.845
DOVER-labeled	0.877	0.869	0.780	0.813	0.870	0.875	0.792	0.829	0.819	0.831	0.843	0.850
Q-Align-labeled	0.878	0.876	**0.794**	0.824	**0.873**	**0.880**	0.781	0.825	0.833	**0.853**	0.848	**0.859**
Students Supervised by Ground Truth Labels (27k)												
LSVQ-labeled	**0.881**	**0.878**	**0.797**	**0.834**	**0.874**	**0.874**	**0.797**	**0.828**	**0.830**	**0.838**	**0.851**	**0.861**
Out of Distribution Datasets	**LIVE-YT-Gaming**		**CGVDS**		**LIVE-YT-HFR**		**Waterloo-IVC-4K**		**KVQ**		**Overall**	
# of videos	600		357		480		1,200		2,926		-	
Methods	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**
Weak Teachers												
MinimalisticVQA(VII)	0.596	0.682	0.681	0.733	0.061	0.130	0.275	0.338	0.604	0.659	0.490	0.551
MinimalisticVQA(IX)	**0.686**	**0.746**	**0.797**	**0.816**	0.301	0.388	**0.459**	**0.502**	**0.615**	**0.661**	**0.574**	**0.622**
FAST-VQA	0.631	0.677	0.725	0.747	0.326	0.415	0.327	0.363	0.518	0.526	0.486	0.512
DOVER	0.647	0.728	0.694	0.747	**0.360**	**0.465**	0.368	0.418	0.559	0.593	0.519	0.569
Q-Align	0.611	0.681	0.756	0.798	0.329	0.342	0.414	0.497	0.613	0.655	0.555	0.606
Students Supervised by Weak Teacher Labels (27k)												
MinimalisticVQA(VII)	0.616	0.713	0.698	0.769	0.355	0.420	0.329	0.384	0.575	0.628	0.515	0.576
MinimalisticVQA(IX)	**0.671**	**0.739**	0.741	0.792	**0.463**	**0.532**	0.440	0.486	**0.623**	**0.669**	**0.582**	**0.633**
FAST-VQA	0.591	0.701	0.722	0.774	0.432	0.479	0.423	0.478	0.578	0.623	0.543	0.597
DOVER	0.636	0.726	**0.760**	**0.813**	0.438	0.516	0.444	0.523	0.567	0.615	0.549	0.611
Q-Align	0.665	0.729	0.744	0.796	0.424	0.481	**0.447**	**0.524**	0.618	0.661	0.578	0.632
Students Supervised by Weak Teacher Labels (200k)												
MinimalisticVQA(VII)-labeled	0.632	0.717	0.718	0.773	0.318	0.386	0.356	0.412	0.604	0.652	0.536	0.593
MinimalisticVQA(IX)-labeled	**0.687**	0.748	**0.763**	**0.810**	0.383	0.461	**0.459**	0.515	**0.638**	**0.676**	**0.591**	**0.639**
FAST-VQA-labeled	0.658	**0.766**	0.752	0.785	0.392	0.422	0.414	0.493	0.585	0.624	0.550	0.604
DOVER-labeled	0.662	0.758	0.752	0.809	**0.449**	**0.482**	0.435	0.519	0.574	0.627	0.554	0.617
Q-Align-labeled	0.671	0.738	0.744	0.785	0.437	0.480	0.450	**0.525**	0.620	0.668	0.581	0.636
Students Supervised by Ground Truth Labels (27k)												
LSVQ-labeled	**0.643**	**0.713**	**0.713**	**0.770**	**0.451**	**0.490**	**0.451**	**0.485**	**0.619**	**0.636**	**0.577**	**0.608**

First, we describe how to construct the probability distribution for comparative quality assessments. The comparative token set is defined as:

$$\mathcal{S} = \{s_k\}_{k=1}^{5} = \{\textit{inferior}, \textit{worse}, \textit{similar}, \textit{better}, \textit{superior}\}. \tag{27}$$

The probability of each token is computed using the softmax function:

$$q_{s_k} = \frac{e^{s_k}}{\sum_{m=1}^{r} e^{s_m}}, \tag{28}$$

where q_{s_k} represents the probability of the k-th token, and r denotes the number of levels.

To obtain a quality score for the test video v_{eval}, we aggregate its comparative probabilities against anchor videos using a weighted summation:

$$P\left(v_{\text{anchor}}, v_{\text{eval}}\right) = \sum_{k=1}^{r} \alpha_k q_{s_k}\left(v_{\text{anchor}}, v_{\text{eval}}\right), \quad r = 1 \ldots p. \tag{29}$$

where α_k are fixed weights that reflect the comparative lev-

Table 5. Performance of our weak-to-strong methods on other state-of-the-art LMMs.

In-domain Datasets	**$\text{LSVQ}_{\text{test}}$**		**$\text{LSVQ}_{\text{1080p}}$**		**KoNViD-1k**		**LIVE-VQC**		**YouTube-UGC**		**Overall**	
# of videos	7,182		3,573		1,200		585		1,020		-	
Methods	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**
Our Weak-to-Strong Methods (Qwen2.5-VL-7B)												
(I): Single teacher supervision *as baseline*	0.877	0.876	0.784	0.820	0.868	0.875	0.783	0.822	0.841	0.840	0.845	0.856
(II): (I) + Ensembling homogeneous teachers	0.879	0.882	0.795	0.827	0.872	0.878	0.793	0.831	0.839	0.841	0.850	0.862
(III): (II) + Integrating heterogeneous teachers	0.882	0.881	0.796	0.829	0.876	0.882	0.791	0.830	0.843	0.844	0.852	0.862
(IV): (III) + Confidence loss	0.883	0.883	0.796	0.830	0.877	0.881	0.789	0.826	0.847	0.849	0.853	0.864
(V): (IV) + Iterative stage W2S training	0.884	0.885	0.797	0.831	0.881	0.878	0.792	0.830	**0.852**	**0.856**	0.854	0.866
(VI): (V) + Iterative stage W2S training	**0.889**	**0.888**	**0.801**	**0.832**	**0.885**	**0.884**	**0.796**	**0.834**	0.848	0.853	**0.858**	**0.868**
Our Weak-to-Strong Methods (InternVL3-8B)												
(I): Single teacher supervision *as baseline*	0.874	0.876	0.789	0.826	0.865	0.872	0.771	0.817	0.832	0.836	0.843	0.857
(II): (I) + Ensembling homogeneous teachers	0.877	0.876	0.796	0.829	0.869	0.875	0.780	0.827	0.834	0.839	0.848	0.859
(III): (II) + Integrating heterogeneous teachers	0.879	0.878	0.796	0.831	0.872	0.877	0.778	0.826	0.839	0.842	0.849	0.861
(IV): (III) + Confidence loss	0.879	0.880	0.794	0.826	0.871	0.876	0.776	0.824	0.842	0.845	0.849	0.860
(V): (IV) + Iterative stage W2S training	0.881	0.881	0.793	0.826	0.873	0.873	0.779	0.817	0.844	0.851	0.850	0.861
(VI): (V) + Iterative stage W2S training	**0.884**	**0.883**	**0.798**	**0.832**	**0.881**	**0.878**	**0.781**	**0.828**	**0.846**	**0.852**	**0.854**	**0.864**
Out of Distribution Datasets	**LIVE-YT-Gaming**		**CGVDS**		**LIVE-YT-HFR**		**Waterloo-IVC-4K**		**KVQ**		**Overall**	
# of videos	600		357		480		1,200		2,926		-	
Methods	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**	**SRCC**	**PLCC**
Our Weak-to-Strong Methods (Qwen2.5-VL-7B)												
(I) - *as baseline*: Single teacher supervision	0.692	0.751	0.774	0.809	0.439	0.516	0.448	0.535	0.647	0.670	0.599	0.645
(II): (I) + Ensembling homogeneous teachers	0.710	0.763	0.779	0.809	0.436	0.501	0.437	0.504	0.662	0.692	0.607	0.650
(III): (II) + Integrating heterogeneous teachers	0.718	0.771	0.784	0.816	0.472	0.538	0.492	0.555	0.697	0.724	0.641	0.682
(IV): (III) + Confidence loss	0.721	0.772	0.782	0.813	0.511	0.587	0.524	0.593	0.719	0.734	0.663	0.700
(V): (IV) + Iterative stage W2S training	0.723	0.776	**0.787**	**0.818**	0.578	0.635	0.590	0.649	0.739	0.757	0.694	0.729
(VI): (V) + Iterative stage W2S training	**0.730**	**0.783**	0.783	0.815	**0.629**	**0.710**	**0.649**	**0.691**	**0.745**	**0.763**	**0.715**	**0.748**
Our Weak-to-Strong Methods (InternVL3-8B)												
(I) - *as baseline*: Single teacher supervision	0.671	0.722	0.758	0.805	0.382	0.459	0.442	0.517	0.633	0.664	0.582	0.630
(II): (I) + Ensembling homogeneous teachers	0.682	0.733	0.762	0.809	0.428	0.503	0.427	0.491	0.651	0.683	0.594	0.640
(III): (II) + Integrating heterogeneous teachers	0.692	0.743	0.774	0.817	0.460	0.531	0.484	0.542	0.683	0.718	0.628	0.673
(IV): (III) + Confidence loss	0.694	0.745	0.772	0.816	0.492	0.570	0.514	0.587	0.704	0.732	0.648	0.694
(V): (IV) + Iterative stage W2S training	0.701	0.752	0.779	0.817	0.569	0.630	0.570	0.616	0.722	0.759	0.677	0.720
(VI): (V) + Iterative stage W2S training	**0.719**	**0.768**	**0.784**	**0.821**	**0.595**	**0.691**	**0.613**	**0.663**	**0.739**	**0.773**	**0.700**	**0.739**

els. Specifically, the weights are defined as:

$$\{\alpha_k\}_{k=1}^{5} = \{0, 0.25, 0.5, 0.75, 1\}. \tag{30}$$

This approach enables the model to generate a continuous quality score for a single video by leveraging its relative comparisons against anchor videos in the training set.

C.3.2. Score Modeling

Finally, we construct a probability matrix based on pairwise comparisons with a set of anchor videos. Given a set of five anchor videos, we first define a probability matrix:

$$M_r \in \mathbb{R}^{5\times5}, \tag{31}$$

where each entry $P(b^{(i)}, b^{(j)})$ represents the probability that anchor video $b^{(i)}$ is preferred over $b^{(j)}$. This probability satisfies:

$$P(b^{(i)}, b^{(j)}) = 1 - P(b^{(j)}, b^{(i)}), \quad P(b^{(i)}, b^{(i)}) = 0.5. \tag{32}$$

To evaluate a test video v_{test}, we compute its comparative probabilities against all anchor videos, forming the probability vector:

$$c = \left[P(b^{(1)}, v_{\text{test}}), P(b^{(2)}, v_{\text{test}}), \ldots, P(b^{(5)}, v_{\text{test}})\right]. \tag{33}$$

Next, we integrate this vector into the complete probability matrix:

$$M \in \mathbb{R}^{(5+1)\times(5+1)}, M = \begin{bmatrix} M_r & c \\ (1-c)^\top & 0.5 \end{bmatrix}. \tag{34}$$

With this probability matrix, we estimate the final quality score using maximum a posteriori (MAP) [21] estimation under Thurstone's Case V model [20]. This is formulated as the following convex optimization problem:

$$\arg\max_{\hat{q}} \sum_{i,j} M_{i,j} \log\left(\Phi(\hat{q}^{(i)} - \hat{q}^{(j)})\right) - \sum_{i} \frac{\hat{q}^{(i)}}{2}, \quad \text{s.t.} \sum_{i} \hat{q}^{(i)} = 0. \tag{35}$$

Here, $\Phi(\cdot)$ denotes the standard normal cumulative distribution function, and the final score $\hat{q}^{(n+1)}$ corresponds to the estimated quality of the test video.

D. More Details of Experimental Results

D.1. More Details of Weak-to-strong Generalization Effect

Table 4 presents the per-dataset results from the experiments described in the main paper (Section 3.3). For in-domain benchmarks, the student model achieves performance comparable to its teachers, and for OOD benchmarks, the student model shows substantial improvements over its teachers, highlighting a pronounced weak-to-strong generalization effect.

D.2. More Details of Performance under Other LMM Backbones

Table 5 presents the performance of our weak-to-strong training paradigm on two additional state-of-the-art LMMs: `Qwen2.5-VL-7B` [1] and `InternVL3-8B` [28]. Although their results are slightly lower than that of `LLaVA-OneVision-Chat-7B` [8] reported in the main paper, the progressively enhanced training pipeline—including unifying diverse supervision signals and applying our iterative weak-to-strong strategies—substantially boosts their performance. Both models ultimately achieve strong results, demonstrating the generality and effectiveness of our proposed training paradigm across different LMM backbones.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 11

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 6

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[4] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019. 6

[5] David Hasler and Sabine E Suesstrunk. Measuring colorfulness in natural images. In *Human vision and electronic imaging VIII*, pages 87–95. SPIE, 2003. 2, 4

[6] Vlad Hosu, Franz Hahn, Mohsen Jenadeleh, Hanhe Lin, Hui Men, Tamás Szirányi, Shujun Li, and Dietmar Saupe. The konstanz natural video database (konvid-1k). In *2017 Ninth international Conference on Quality of Multimedia experience*, pages 1–6, 2017. 6

[7] ITU-T P.910. Subjective video quality assessment methods for multimedia applications, 2008. 2

[8] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 11

[9] Zhuoran Li, Zhengfang Duanmu, Wentao Liu, and Zhou Wang. Avc, hevc, vp9, avs2 or av1?—a comparative study of state-of-the-art video encoders on 4k videos. In *Image Analysis and Recognition: 16th International Conference, ICIAR 2019, Waterloo, ON, Canada, August 27–29, 2019, Proceedings, Part I 16*, pages 162–173. Springer, 2019. 6

[10] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022. 6

[11] Yiting Lu, Xin Li, Yajing Pei, Kun Yuan, Qizhi Xie, Yunpeng Qu, Ming Sun, Chao Zhou, and Zhibo Chen. Kvq: Kwai video quality assessment for short-form videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25963–25973, 2024. 6

[12] Pavan C Madhusudana, Xiangxu Yu, Neil Birkbeck, Yilin Wang, Balu Adsumilli, and Alan C Bovik. Subjective and objective quality assessment of high frame rate videos. *IEEE Access*, 9:108069–108082, 2021. 6

[13] Niranjan D Narvekar and Lina J Karam. A no-reference image blur metric based on the cumulative probability of blur detection (cpbd). *IEEE Transactions on Image Processing*, 20(9):2678–2683, 2011. 2

[14] Juergen Pandel. Measuring of flickering artifacts in predictive coded video sequences. In *2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, pages 231–234. IEEE, 2008. 2, 3

[15] Eli Peli. Contrast in complex images. *JOSA A*, 7(10):2032–2040, 1990. 2

[16] Piotr Romaniak, Lucjan Janowski, Mikolaj Leszczuk, and Zdzislaw Papir. Perceptual quality assessment for h. 264/avc compression. In *2012 IEEE Consumer Communications and Networking Conference*, pages 597–602. IEEE, 2012. 2

[17] Avinab Saha, Yu-Chih Chen, Chase Davis, Bo Qiu, Xiaoming Wang, Rahul Gowda, Ioannis Katsavounidis, and Alan C Bovik. Study of subjective and objective quality assessment of mobile cloud gaming videos. *IEEE Transactions on Image Processing*, 32:3295–3310, 2023. 6

[18] Zeina Sinno and Alan Conrad Bovik. Large-scale study of perceptual video quality. *IEEE Transactions on Image Processing*, 28(2):612–627, 2018. 6

[19] Wei Sun, Wen Wen, Xiongkuo Min, Long Lan, Guangtao Zhai, and Kede Ma. Analysis of video quality datasets via

design of minimalistic video quality models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 6

[20] Louis L Thurstone. A law of comparative judgment. In *Scaling*, pages 81–92. Routledge, 2017. 10

[21] Kristi Tsukida, Maya R Gupta, et al. How to analyze paired comparison data. *Department of Electrical Engineering University of Washington, Tech. Rep. UWEETR-2011-0004*, 1, 2011. 10

[22] Yilin Wang, Sasi Inguva, and Balu Adsumilli. Youtube ugc dataset for video compression research. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing*, pages 1–5. IEEE, 2019. 6

[23] Haoning Wu, Chaofeng Chen, Jingwen Hou, Liang Liao, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Fast-vqa: Efficient end-to-end video quality assessment with fragment sampling. In *European Conference on Computer Vision*, pages 538–554. Springer, 2022. 6

[24] Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20144–20154, 2023. 6

[25] Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, et al. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. *arXiv preprint arXiv:2312.17090*, 2023. 6

[26] Zhenqiang Ying, Maniratnam Mandal, Deepti Ghadiyaram, and Alan Bovik. Patch-vq:'patching up'the video quality problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14019–14029, 2021. 6

[27] Xiangxu Yu, Zhengzhong Tu, Zhenqiang Ying, Alan C Bovik, Neil Birkbeck, Yilin Wang, and Balu Adsumilli. Subjective quality assessment of user-generated content gaming videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 74–83, 2022. 6

[28] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 11