Appendix

Supplementary Materials

A. Proofs

In [5], the inputs to the parallel streams are learnable transformation of the same input x, so that one can assume that each parallel stream follows (7) in an unbiased way, leading to a simplification in the analysis of the parallel scaling law. We start by reviewing the result from [5].

Lemma 1 ([5]). *The loss of ParScale with J streams, each indicated with p_j, follow*

$$L^{\text{ParScale}}(N, J) = E + \frac{A}{(NJ^{1/\alpha})^\alpha} D + \mathcal{O}(\Delta_j^3), \quad D := (J-1)\rho + 1, \tag{11}$$

where $\Delta_j := \frac{p_j - p}{p}$. D depends on the correlation ρ of the residuals between the streams. The loss decays fastest when $\rho = 0$, and degrades to the original Chinchilla law when $\rho = 1$.

Proof. For each stream, following (7), the CE loss reads

$$\begin{aligned} L_i &= \mathbb{E}_{x,y}[-\log p_j(y|x)] && (12)\\ &= \mathbb{E}_{x,y}[-\log\{p \cdot (1+\Delta_j)\}] && (13)\\ &= \underbrace{\mathbb{E}_{x,y}[-\log p]}_{=:E} + \underbrace{\mathbb{E}_{x,y}[-\log(1+\Delta_j)]}_{A/N^\alpha}, && (14) \end{aligned}$$

which can be deduced by comparing the loss with (7). Using the Taylor expansion $\log(1+x) = x - \frac{x^2}{2} + \mathcal{O}(x^3)$, we can expand the second term of the rhs

$$\mathbb{E}_{x,y}\left[\Delta_j - \frac{\Delta_j^2}{2} + \mathcal{O}(\Delta_j^3)\right] = \mathbb{E}_{x,y}\left[\frac{p_j - p}{p}\right] + \mathbb{E}_{x,y}\left[\frac{\Delta_j^2}{2}\right] + \mathcal{O}(\Delta_j^3). \tag{15}$$

Since $\mathbb{E}_{x,y}[p_j] = p$, we have that

$$\mathbb{E}_{x,y}\left[\Delta_j^2\right] = 2A/N^\alpha + \mathcal{O}(\Delta_j^3). \tag{16}$$

Now, we are ready to compute the loss L by averaging the streams. Let $\Delta := \frac{1}{J}\sum_{j=1}^J \Delta_j$. The loss reads

$$\begin{aligned} L^{\text{ParScale}}(N, J) &= E + \mathbb{E}_{x,y}[-\log(1+\Delta)] && (17)\\ &= E + \mathbb{E}_{x,y}\left[\frac{\Delta^2}{2}\right] + \mathcal{O}(\Delta^3) && (18)\\ &= E + \frac{1}{2J^2}\mathbb{E}_{x,y}\left[\sum_{j=1}^J \Delta_j^2 + 2\sum_{j<k}\Delta_j\Delta_k\right] + \mathcal{O}(\Delta_j^3) && (19)\\ &\overset{(16)}{=} E + \frac{A}{JN^\alpha} + \frac{1}{J^2}\mathbb{E}_{x,y}\left[\sum_{j<k}\Delta_j\Delta_k\right] + \mathcal{O}(\Delta_j^3) && (20)\\ &= E + \frac{A}{JN^\alpha} + \frac{1}{J^2}\cdot J(J-1)\cdot\rho\frac{A}{N^\alpha} + \mathcal{O}(\Delta_j^3) && (21)\\ &= E + \frac{A}{(NJ^{1/\alpha})^\alpha}\left[(J-1)\rho + 1\right] + \mathcal{O}(\Delta_j^3) && (22) \end{aligned}$$

□

We are now ready to derive our proposition.

Proposition 1. *A VideoLLM that is shown some subset of frames x_j follow*

$$L_j^{\text{VideoLLM}}(N) = E + \frac{A}{N^\alpha} + B_j + \mathcal{O}(B_j^2) + \mathcal{O}(\Delta_j^3). \tag{23}$$

Further, the expected CE loss of VPS with J streams follow

$$L^{VPS}(N, J) = E + \frac{A}{(NJ^{1/\alpha})^\alpha}[1 + (J-1)\rho] + \bar{B}(J) + \mathcal{O}(\bar{B}(J)^2) + \mathcal{O}(\Delta_j^3), \tag{24}$$

where $\bar{B}(J) = \frac{1}{J}\sum_{j=1}^{J} B_j$, and ρ is the correlation coefficient between $\varepsilon_i, \varepsilon_j, i \neq j$.

Proof. First, recall the definition $B_j := -\mathbb{E}_{x,y}[\Delta_j]$, $\varepsilon_j := \Delta_j + B_j$. The per-stream cross-entropy is

$$L_j^{\text{VideoLLM}}(N) = -\mathbb{E}_{x,y}[\log p_j] = E + \mathbb{E}_{x,y}[-\log(1 + \Delta_j)]. \tag{25}$$

In order for us to use Taylor approximation, first note that

$$\mathbb{E}_{x,y}[\Delta_j] = \mathbb{E}_{x,y}[\varepsilon_j - B_j] = -B_j, \tag{26}$$

as B_j is a constant w.r.t. the expectation in $p(x, y)$. Further, we have that

$$\begin{aligned}
\mathbb{E}_{x,y}[\Delta_j^2] &= \mathbb{E}_{x,y}\left[(\varepsilon_j - B_j)^2\right] && (27)\\
&= \mathbb{E}_{x,y}[\varepsilon_j^2] - 2\mathbb{E}_{x,y}[\varepsilon_j B_j] + \mathcal{O}(B_j^2) && (28)\\
&= \mathbb{E}_{x,y}[\varepsilon_j^2] + \mathcal{O}(B_j^2) \quad (\because \mathbb{E}[\varepsilon_j] = 0) && (29)
\end{aligned}$$

Substituting these back into the loss equation leads to

$$\begin{aligned}
L_j^{\text{VideoLLM}}(N) &= E + \frac{1}{2}\mathbb{E}[\varepsilon_j^2] + B_j + \mathcal{O}(B_j^2) + \mathcal{O}(\Delta_j^3) && (30)\\
&\overset{(16)}{=} E + \frac{A}{N^\alpha} + B_j + \mathcal{O}(B_j^2) + \mathcal{O}(\Delta_j^3), && (31)
\end{aligned}$$

where we used the result from Lemma 1 to substitute for the variance of the residual. We have completed the first part of the proof.

For VPS, let $\bar{\varepsilon} := \sum_j \varepsilon_j / J$. We have

$$\mathbb{E}_{x,y}[\Delta] = \mathbb{E}_{x,y}[\bar{\varepsilon} - \bar{B}] = -\bar{B} \tag{32}$$

and

$$\begin{aligned}
\mathbb{E}_{x,y}[\Delta^2] &= \frac{1}{J^2}\mathbb{E}\left[\left(\sum_{j=1}^{J}\Delta_j^2\right)\right] && (33)\\
&= \frac{1}{J^2}\mathbb{E}\left[\sum_{j=1}^{J}\Delta_j^2 + 2\sum_{j<k}\Delta_j\Delta_k\right] && (34)\\
&= \frac{1}{J^2}\left(\frac{JA}{N^\alpha} + J\mathcal{O}(B_j^2) + J(J-1)\cdot\rho\sqrt{\mathbb{E}[\Delta_j^2]\mathbb{E}[\Delta_k^2]}\right) && (35)\\
&= \frac{A}{(NJ^{1/\alpha})^\alpha}[1 + (J-1)\rho] + \mathcal{O}(\bar{B}(J)^2) && (36)
\end{aligned}$$

Then, the CE loss for VPS reads

$$\begin{aligned}
L^{VPS}(N, J) &= E + \mathbb{E}_{x,y}[-\log(1 + \Delta)] && (37)\\
&= E - \underbrace{\mathbb{E}_{x,y}[\Delta]}_{\overset{(32)}{=} -\bar{B}} + \frac{1}{2}\mathbb{E}_{x,y}[\Delta^2] + \mathcal{O}(\Delta^3) && (38)\\
&\overset{(36)}{=} E + \frac{A}{(NJ^{1/\alpha})^\alpha}[1 + (J-1)\rho] + \bar{B}(J) + \mathcal{O}(\bar{B}(J)^2) + \mathcal{O}(\Delta^3) && (39)
\end{aligned}$$

□

Table 6. **Results of VPS with logit averaging vs. probability averaging.** Both choices lead to similar results. Small: smallest variant (3B, 2B, 4B) / Base: base variant (7B, 8B, 12B).

			Qwen2.5-VL				InternVL3				Gemma3			
	J	Method	2	4	8	16	2	4	8	16	2	4	8	16
Small	2	logit	0.496	**0.535**	**0.548**	0.572	**0.435**	**0.508**	**0.501**	**0.525**	**0.567**	**0.533**	0.477	0.469
		prob	**0.498**	0.523	0.540	**0.577**	0.433	0.489	0.491	0.523	0.550	0.521	**0.491**	**0.472**
	4	logit	0.498	0.545	**0.577**	0.555	0.420	**0.491**	0.489	0.542	**0.569**	**0.542**	0.491	0.464
		prob	**0.511**	0.545	0.564	**0.596**	**0.423**	0.474	**0.496**	**0.545**	0.548	0.526	**0.513**	**0.486**
Base	2	logit	0.557	0.601	**0.650**	0.670	0.557	0.586	**0.618**	0.633	**0.784**	0.777	0.753	**0.728**
		prob	0.570	**0.633**	0.648	**0.689**	0.557	**0.587**	0.616	0.633	0.782	**0.779**	**0.755**	0.725
	4	logit	0.569	0.623	**0.675**	**0.682**	**0.572**	**0.596**	0.623	0.638	**0.792**	**0.785**	0.753	**0.741**
		prob	0.569	0.623	0.660	0.667	0.565	0.591	0.623	**0.640**	0.790	0.782	**0.759**	0.739

B. Entropy Weighting

Our canonical weighting scheme equally weights the stream. However, in many cases, the different subsets of frames contain varying information—some may contain keyframes that directly answers the given question, while others may be less useful. A natural way to capture whether a given stream is useful is to use entropy

$$H(x) = -\sum_{y} p(y|x) \log p(y|x). \tag{40}$$

When the entropy is high, it means that the output probability of the current token y is close to uniform, which in turn, means that the model is less confident in its answer. In contrast, when the entropy is low, the model is more confident in its answers. Hence, while we do not have oracle access to which subsets of frames are more useful, we can leverage the output entropy as a proxy to measure this value, as also supported by recent literature [25, 39, 47].

We propose an entropy weighting scheme, with

$$w_j(x) \propto \exp\left(-\beta H_j(x)\right), \tag{41}$$

with $\beta \geq 0$ as a hyper-parameter. To ensure normalization, we employ softmax. Setting $\beta = 0$ reduces to our canonical uniform weighting scheme, and larger values of β puts more emphasis on the confident streams. When the model is queried to output a direct answer, as in the benchmark settings, we can additionally use the structure of the problem. Take the multiple-choice question setting of VideoMME, where the model should answer `A/B/C/D`. In such cases, taking the entropy over the whole vocabulary yields noisy estimate of the model entropy. Thus, we further propose to use the *restricted* entropy over the vocabulary of interest[5],

# Frames	**8**		**16**	
J	4	8	4	8
Uniform ($\beta = 0.0$)	0.508	0.528	0.541	0.552
Entropic ($\beta = 7.0$)	**0.534**	**0.560**	**0.563**	**0.565**

Table 7. **VPS with entropic weighting.** Results on Video-MME with Qwen2.5-VL-7B.

$$\bar{H}_j(x) = \sum_{c=1}^{K} p_j(c|x) \log p_j(c|x), \tag{42}$$

where K is the cardinality of the vocabulary space of our interest.

C. Further Results

Probability and logit averaging In Tab. 6, we compare the results of logit averaging and probability averaging when implementing VPS. Across different model classes, we find that both approaches lead to similar results. Thus, while we assume probability averaging in the theoretical analysis for simplicity, we resort to logit averaging for implementation.

Qualitative results We present additional examples of the free form description task in Tab. 9.

[5]For instance, in VideoMME, we consider tokens that are upper or lowercase A/B/C/D with optional spaces.

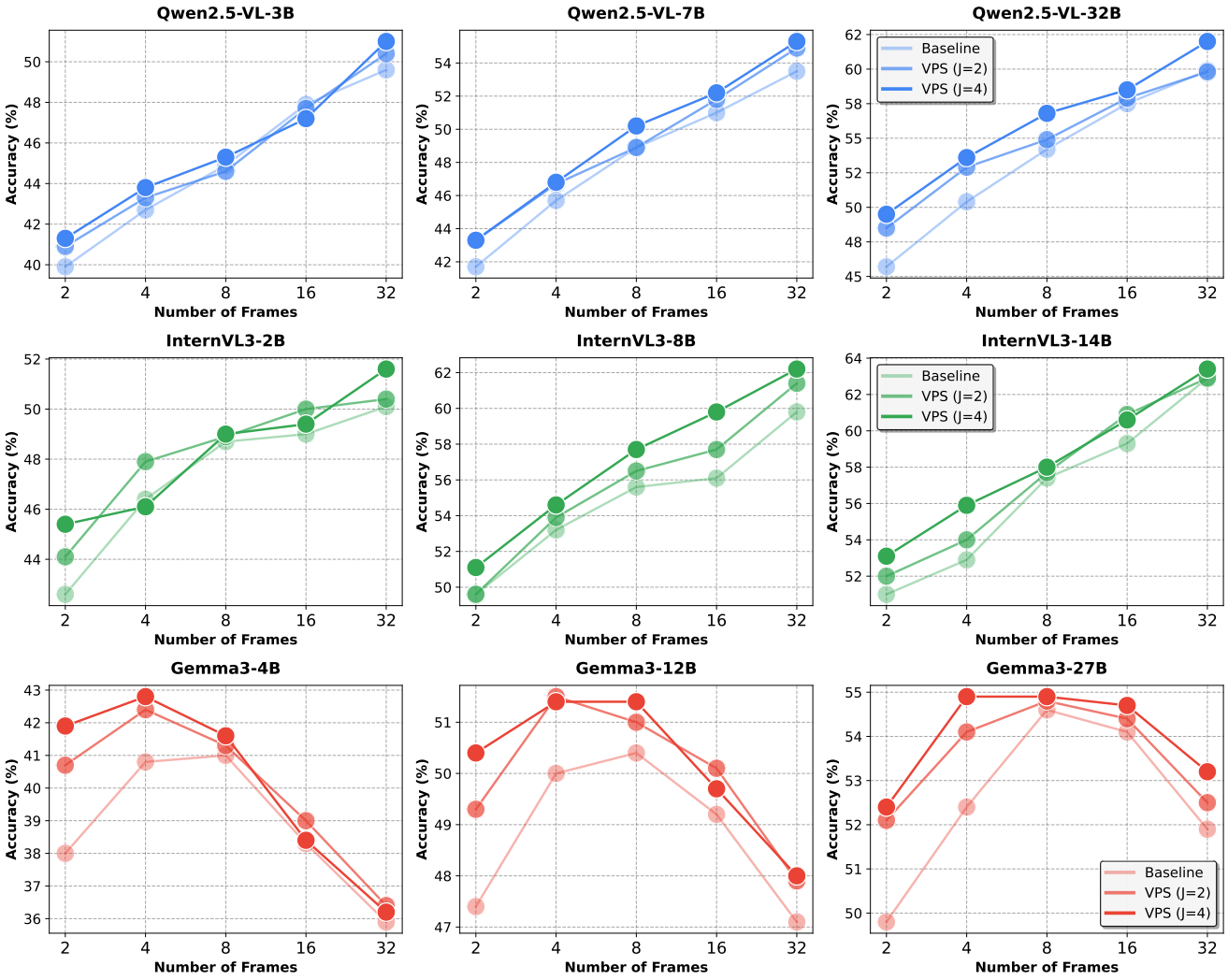

Figure 5. **VPS consistently improves performance across all dimensions**. Across 3 different model classes (Qwen-2.5-VL, InternVL3, Gemma3), 3 different size (2B - 32B), and number of frames used in context. y-axis denotes the accuracy in the Video-MME multiple-choice QA.

D. Experimental Details

D.1. Main experiment

For Video-MME, following each question, we use the following prompt: ``Your response should be a single character: A, B, C, or D. Do not include any other text or explanation.''. For EventHallusion, following each question, we use the following prompt: ``Please answer yes or no.''. All results in the main experiments are obtained through standard sampling with temperature set to 1.0, which is different from the benchmark evaluation settings, where the standard is to resort to greedy decoding.

D.2. Self consistency experiment

Notice that when decoding a single token, majority voting after decoding, and averaging the probabilities before sampling lead to the same results in the limit of infinite samples. In practice, we notice that due to formatting issues, the performance is slightly better when we use majority voting after the answer is extracted, as such scaffolding generally helps. As the goal in this experiment is to emphasize the importance of using different frames for each stream, we also implement VPS with majority voting, but with seeing different frames for each stream, for fair comparison against Self-consistency. This way, the

difference in scaling solely comes from the difference in input frames.

System prompt:
You are an intelligent chatbot designed for evaluating the correctness of generative outputs for video summaries. Your task is to compare the predicted answer with the pseudo-reference answer and determine if they match meaningfully.
You should rely more on the video frames than the pseudo-reference caption.
——
INSTRUCTIONS:
- Focus on the meaningful match between the predicted answer and the pseudo-reference answer.
- Consider synonyms or paraphrases as valid matches.
- Evaluate the correctness of the prediction compared to the video frames.
User prompt:
Given the reference caption, evaluate the quality of the predicted caption.
Provide your evaluation only as an integer value between 1 and 5, with 5 indicating the highest quality.
Please provide your evaluation in the following format: [evaluation]

Table 8. Prompts used for LLM-as-a-judge [62] evaluation system.

D.3. Free form experiment

Since the (pseudo-)ground truth answer in EventHallusion is given in a simple sentence, we use ``Summarize the video in one sentence.'' as the prompt. When using LLM-as-a-judge [62] when evaluating the free form descriptions of the video, we follow [61] and use the prompt specified in Tab. 8.

D.4. Incorporating other strategies

For TCD, we construct a negative stream so that the half the frames are zeroed-out in an interleaved fashion. Let x' be the frame-dropped version of the sub-sampled video. Then, TCD is implemented with

$$\tilde{p}^{\theta}(y|x) = (1+\alpha)p^{\theta}(y|x) - \alpha p^{\theta}(y|x'), \tag{43}$$

where $\alpha \in [0, 1)$ is a constant. Additionally, we set a hyperparameter $\beta \in [0, 1]$ that keeps only the high probability tokens that exceeds the value $\beta \max_w p^{\theta}(y|x)$, following [30]. We use the default values $\alpha = 0.5, \beta = 0.1$. When used together with VPS, we use $\tilde{p}^{\theta}$ instead of p^{θ}. For RITUAL, we consider the following 5 augmentations: horizontal flip, vertical flip, 180 degrees random rotation, color jitter, and gaussian blur. We apply the same augmentation to each frame, and use equal weighting for the original view and the augmented view. When used together with VPS, we construct an augmented view per VPS stream.

D.5. Frame sampling strategy

The dense sampling strategy with J streams first makes J chunks from the video sequence. Within the chunk, the frames are uniformly sampled. BOLT [34] computes the CLIP similarity between the video frames and the query prompt. A normalized score s_i is then sharpened with

$$s_i^r = \left(\frac{s_i - \min(s)}{\max(s) - \min(s)} \right)^{\alpha} \tag{44}$$

, where $\alpha = 3.0$. We then sample the frames according to this sharpened distribution. When using BOLT together with VPS, it is important that we do not sample the same frames for each stream. Hence, we eliminate the frame indices once the frame is sampled from one of the streams, then sample from the truncated distribution.

Video	
Baseline	A cyclist crashes while riding at night, resulting in her bicycle falling over.
VPS ($J = 4$)	A cyclist is struggling to carry her bike down a street.
Video	
Baseline	A parrot curiously interacts with a cup of tea and a spoon, attempting to participate in a human's tea-drinking ritual.
VPS ($J = 4$)	A parrot curiously tries to stir a cup of tea with a spoon.

Table 9. **Qualitative analysis of VPS.** VPS effectively captures the overall motion depicted in videos from the EventHallusion dataset.