DynaStride: Dynamic Stride Windowing with MMCoT for Multi-Scene Captioning

Eddison Pham[1], Prisha Priyadarshini[1], Adrian Maliackel[1], Kanishk Bandi[1], Cristian Meo[1,2,3], Kevin Zhu[1]

Algoverse AI Research[1], LatentWorlds AI[2], Delft University of Technology[3]

Why Scene-Level Captioning?

• Instructional videos are widely used to teach complex tasks through step-by-step guidance. One way is to leverage deep learning models to generate scene-level captions (Narasimhan et al., 2023; Shi and Ji, 2019).

• The growth of AI/ML, particularly in LLMs and VLMs has allowed these scene captioning to be more effective in understanding visual cues and temporal progression (Elstad, 2024; Morales-Navarro and Kafai, 2024)

• Captioning videos improves accessibility for visually impaired, efficiency in indexing and content summarization (Gernsbacher, 2015).

• Recent empirical studies show that inclusion of automated captioning educational videos improve video comprehension, satisfaction, and listening performance (Malakul and Park, 2023; Alabsi, 2020).

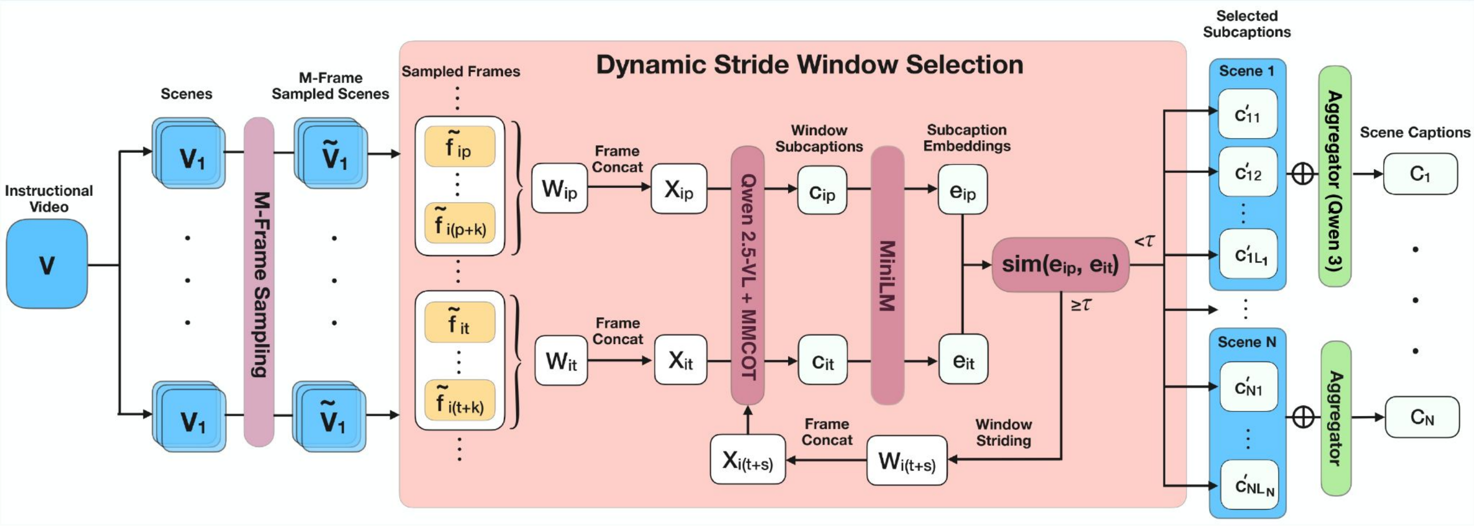
Limitations of current captioning approaches

- **Redundancy vs. brevity:** Dense captions include irrelevant details, while shorter captions miss key actions or temporal relations (Chai et al., 2024; Yang et al., 2023; Tang et al., 2025).
- **Scalability and reproducibility:** Reliance on localized inference tools limits performance on long or complex instructional videos (Chai et al., 2024).
- **Need for context-aware modeling:** Current methods struggle to capture essential actions, objects, and correct event sequences.

High-level Overview of Our Methodology

(1) Sampling and windowing frames in each scene
(2) Leveraging VLM + MMCoT to generate subcaptions
(3) Dynamic stride windowing to skip content-redundant windows
(4) Subcaption aggregation

(2) and (3) works together



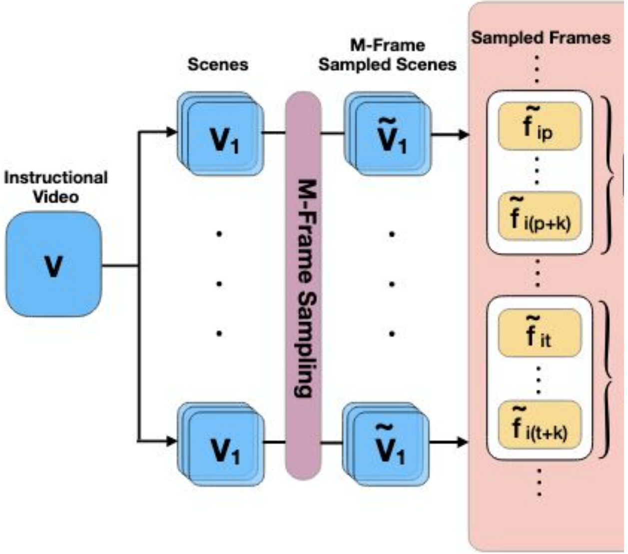
Step 1: Frame Sampling

Scene as frame sequence: Each scene is represented as an ordered sequence of frames.

Subsampling for efficiency: Only every M-*th* frame (specified in paper) is selected to reduce computational cost.

Sliding windows: K-sized window of frames capture short-term temporal dynamics.

Focus on local patterns: Windowing allows precise feature extraction while avoiding processing similar frames unnecessarily.

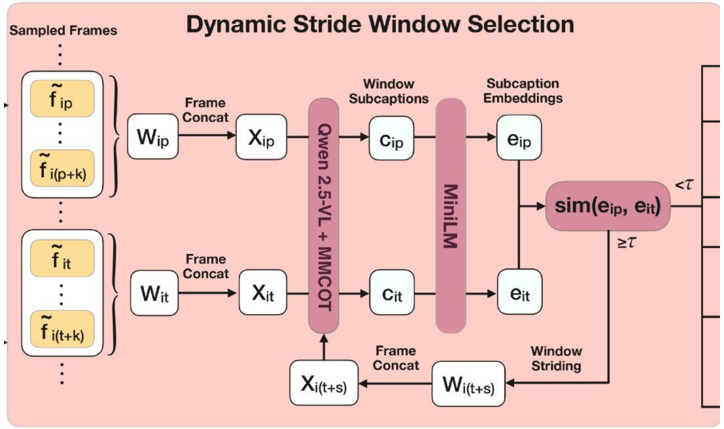


Step 2: MMCoT Subcaptions

Temporal context via wide-frame input: Concatenate frames within each window into a single wide image so the model sees multiple frames at once.

Subcaption generation: We leverage Qwen3 to generate action-objects description pairs of the form "[action] | [objects]" for the current and candidate window.

Multimodal CoT: Encourages the model to understand both temporal dynamics and semantic content, reducing the extraction of irrelevant actions or objects by leveraging local context within each window.

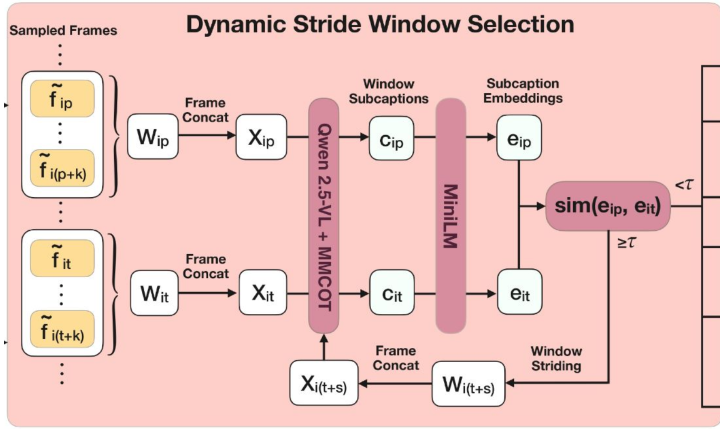


Step 3: Dynamic Stride Window Selection

Embedding-based comparison: Compute embeddings of subcaptions with *MiniLM* embedder.

Similarity threshold: If a candidate window is too similar to the previous one, it is skipped to avoid redundancy.

Dynamic stride: After skipping high similarity windows we scale the stride for upcoming windows. Repeat until end of video or dissimilarity detected → reset the stride scaling variable.

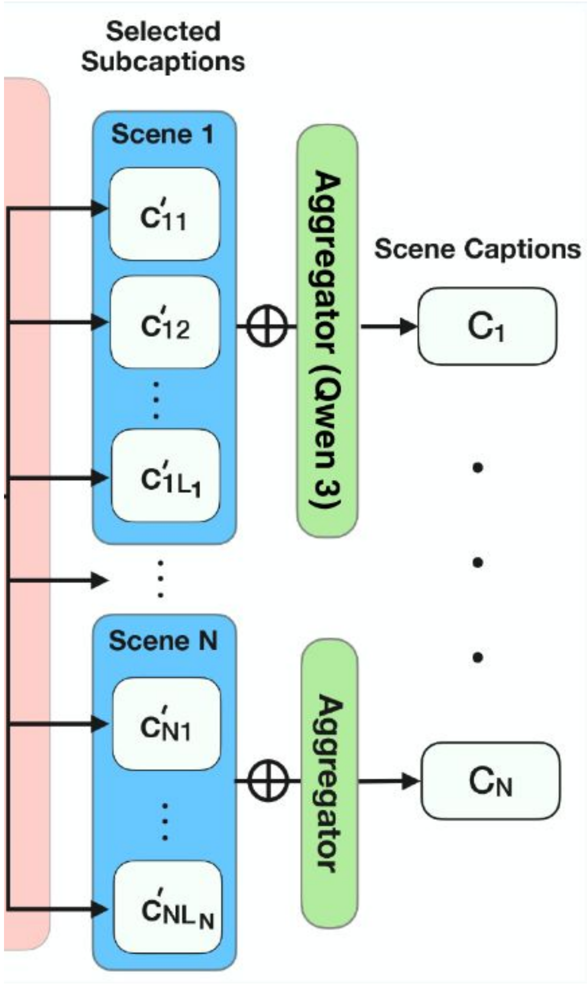


Step 4: Subcaption Aggregation

The first selected subcaption per scene is the first window. Every subcaption afterwards are chosen via the Dynamic Stride Window algorithm.

Combine retained subcaptions: Concatenate selected subcaptions into a single input for the aggregation model.

Generate final caption: Use a Qwen3-4B-Instruct model to produce a coherent instructional caption for the entire scene.



Experiments and Datasets

- **Dataset:** YouCookII (uniformly sampled 210 videos from validation set)
- **No Training/Fine-tuning Involved:** The pipeline solely leverages pretrained models
- **Baselines:** GPT-4o, Video-LLaMA3
- **Subcaption Aggregators:** Qwen3, Phi-3, Mistral
- **Metrics:** BLEU-4, METEOR, CIDEr, BERTScore, SBERT (3 seeds)

Research Questions:

- RQ 1: To what extent does our method improve the coherence and meaningfulness of the inferred scene captions?
- RQ 2: How does the frame sampling and aggregator impact overall caption quality?

Main Experiment

Method	N-gram			BERT			SBERT
	B@4(↑)	METEOR(↑)	CIDEr(↑)	Prec(↑)	Recall(↑)	F1(↑)	(Sim↑)
GPT-4o	**4.73**(0.63)	**28.47**(1.37)	0.48(0.06)	0.19(0.01)	**0.29**(0.01)	0.23(0.01)	0.60(0.01)
VLLaMA-3	4.10(0.32)	22.71(0.63)	0.49(0.02)	0.19(0.01)	0.22(0.00)	0.21(0.01)	0.58(0.00)
DynaStride	4.18(0.07)	24.31(0.10)	**0.56**(0.00)	**0.25**(0.00)	0.26(0.00)	**0.27**(0.00)	**0.61**(0.00)

Table 1: Scene captioning results on YouCook2 validation set, comparing GPT-4o, LLaMA-3, and our method.

DynaStride achieves the highest CIDEr and semantic metrics compared to baselines.

- Outperforms GPT-4o in CIDEr, BERT Precision, BERT F1, and SBERT
- Outperforms VLLaMA-3 in **ALL** metrics.

Ablation Results

Configurations	N-gram			BERT			SBERT
	B@4(↑)	METEOR(↑)	CIDEr(↑)	Prec(↑)	Recall(↑)	F1(↑)	Sim(↑)
Aggregator Comparison							
Phi	2.78 (0.24)	22.8 (0.38)	0.37 (0.02)	0.17 (0.01)	0.25 (0.00)	0.21 (0.01)	0.59 (0.00)
Mistral	3.36 (0.08)	19.49 (0.12)	0.51 (0.01)	**0.27** (0.00)	0.23 (0.00)	0.25 (0.00)	0.60 (0.00)
Qwen3	**4.18** (0.07)	**24.31** (0.10)	**0.56** (0.00)	0.25 (0.00)	**0.26** (0.00)	**0.27** (0.00)	**0.61** (0.00)
Frame Sampling Rates							
GPT-4o							
5 Frames	4.31 (0.15)	27.58 (0.47)	0.45 (0.02)	0.18 (0.00)	0.28 (0.00)	0.23 (0.00)	0.59 (0.00)
20 Frames	4.69 (0.19)	27.91 (0.28)	0.49 (0.02)	0.19 (0.01)	0.29 (0.00)	0.24 (0.00)	0.60 (0.00)
40 Frames	4.52 (0.11)	**28.07** (0.03)	0.48 (0.00)	0.19 (0.00)	**0.29** (0.00)	0.24 (0.00)	0.60 (0.00)
VLLaMA-3							
5 Frames	3.60 (0.45)	22.48 (0.17)	0.45 (0.05)	0.18 (0.00)	0.22 (0.00)	0.19 (0.00)	0.57 (0.00)
20 Frames	4.32 (0.29)	22.28 (0.27)	0.52 (0.02)	0.22 (0.00)	0.22 (0.00)	0.22 (0.00)	0.58 (0.00)
40 Frames	4.79 (0.05)	21.90 (0.08)	0.56 (0.01)	**0.27** (0.00)	0.21 (0.00)	0.24 (0.00)	0.58 (0.00)
DynaStride							
5 Frames	3.89 (0.12)	23.38 (0.14)	0.52 (0.01)	0.24 (0.00)	0.25 (0.00)	0.24 (0.00)	0.60 (0.00)
20 Frames	4.48 (0.03)	24.82 (0.10)	0.58 (0.00)	0.24 (0.00)	0.26 (0.00)	0.25 (0.00)	0.61 (0.00)
40 Frames	**4.91** (0.03)	26.36 (0.18)	**0.61** (0.00)	0.25 (0.00)	0.28 (0.00)	**0.27** (0.00)	**0.63** (0.00)

Sparse sampling boosts caption quality and aggregator choice impacts consistency.
- DynaStride benefits from sparser sampling, with 20–40 frames yielding the highest CIDEr, F1, and SBERT similarity.
- Qwen3 produces the most consistent and accurate captions, while other aggregators like Phi show much higher variability.

Limitations

• **Dependence on pretrained models**: Reliance on pre-train models may limit generalization beyond the YouCookII domain.

• **Dataset constraints**: YouCookII is relatively small and may not represent the full diversity of instructional tasks, limiting applicability to other domains or complex workflows.

• **Dynamic frame sampling trade-offs**: While efficient, it may miss subtle or rapid actions, producing incomplete, ambiguous, or temporally inconsistent subcaptions.

• **Subcaption aggregation issues**: The dynamic stride algorithm reduces redundancy but may not fully prevent coherence or clarity issues in the final scene-level captions.

• **Lack of adaptation or feedback mechanisms**: No domain adaptation or human feedback is incorporated, limiting continuous improvement or personalization for diverse learners.

Possible Future Directions

- Extend DynaStride to raw, unsegmented videos using robust scene boundary detection (e.g., temporal action detection, weakly supervised segmentation).
- Expand to diverse instructional domains beyond YouCookII for broader generalization.
- Experiment with fine-tuning the VLM or Aggregator models to better align the captions to domain specific tasks.
- Incorporate human evaluations to assess practical usefulness and educational impact.

Thank you for Listening!

DynaStride Paper