VINAYA: YOKED SCHEDULES FOR STREAMING NON-AUTOREGRESSIVE CAPTIONING

Anonymous authors
Paper under double-blind review

ABSTRACT

A streaming captioner must decide how much of a clip it has seen before writing every word. Autoregressive models answer this by construction at the cost of one sequential pass per token. Non-autoregressive models are faster but cannot stream because no prefix is guaranteed final until the last pass and nothing can leave early. We introduce ViNAYA (Vision Non-Autoregressive Yoked Architecture), which closes this gap by turning the write order into a declared schedule rather than a learned behaviour. A monotone text frontier commits and emits spans from left to right, while a yoked monotone source horizon reveals the clip at a matching pace. Both frontiers are closed mathematical forms fixed before decoding begins. Therefore, no emitted token can depend on future source context, and every emitted span is final the moment it leaves. ViNAYA's two-knob design permits separate control over both the source reveal and the text schedule, with the two knobs joined by a ratio ν that has a closed-form rule for choosing it from the corpus geometry. This coupling yields a predictable exposure budget of $1/(1+\nu)$, which we validate against measured exposure on three video captioning corpora with three frozen encoders. On the same corpora, ViNAYA reduces Average Lagging by 45 to 50 percent against the strongest NAR streaming baseline, at quality at or above parity in most cells and with core architectural elements shared by construction. While an autoregressive wait-k baseline retains a small quality edge on two corpora, ViNAYA compensates for that with up to 3 times fewer forward passes per window and a sequential cost that stays bounded as captions grow. The per-token structure of autoregression forces its inference time to grow with output length, and on ActivityNet, where captions are longest, the autoregressive baseline requires roughly 427 seconds of test-split inference against 179 seconds for ViNAYA.

1 INTRODUCTION

Non-autoregressive decoding sits at the intersection of the two dominant ideas in modern sequence modelling, full attention (Vaswani et al., 2017) and iterative denoising (Austin et al., 2021; Li et al., 2022). It combines them by allowing every output position to see every other position, and by refining a stochastic masked canvas over a handful of parallel passes. The result is a decoder that writes many tokens at once instead of one at a time.

Autoregressive systems remain the industry standard despite this speed advantage. They are simple, they match quality, and they stream naturally, because writing one token per step means a prefix is always ready to emit. Non-autoregressive systems have seen slower adoption because their iteration rule is based on content rather than position. No prefix is certified as final until the last pass, and a mask-predict decoder (Ghazvininejad et al., 2019) cannot stream, because nothing can leave early with certainty.

The two properties we need, parallel writing and monotone commitment, have not been combined before. Every existing streaming captioner is autoregressive by construction, and every existing parallel decoder leaves the write order to be discovered rather than declared. We close this gap by building, to our knowledge, the first streaming multimodal non-autoregressive captioner whose read and write schedule is fixed in closed form before decoding begins. A text frontier commits and emits segments from left to right. A yoked source horizon reveals the clip at the matching pace. Because

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

both frontiers are declared in advance, the schedule is a closed-form expression whose parameters follow from corpus geometry rather than a policy that has to be learned.

The single-exponent schedule is already established in the autoregressive streaming setting, where it reduces context dilution and enforces a dependency guarantee (Mehmood & Javed, 2026). A natural first attempt is to reuse it here by driving both frontiers with that same exponent, but this collapses the expected fraction of source visible per emitted token to one half for every value, and the dial loses control over the quantity it was built to set (Theorem 1, Appendix B). Two exponents restore control. We write them as γ_y for the text frontier and γ_v for the source horizon, and the expected exposure depends only on their ratio, $\nu = \gamma_v/\gamma_y$. For our headline experiments we hold ν fixed at 0.5 and vary it separately in Section 4 to isolate its effect. In Appendix B we provide a rule for choosing ν from corpus geometry. Therefore, the text exponent γ_y is the only dial that needs sweeping, and the source exponent follows automatically as: $\nu \cdot \gamma_y$.

We evaluate ViNAYA on three video captioning corpora with three frozen visual encoders, spanning both regimes of the source-to-text geometry: YouCook2 and Charades-STA are source-heavy, while ActivityNet Captions is text-heavy, with a sparser visual source supporting a longer caption. As no off-the-shelf streaming baseline exists for this setting, we compare ViNAYA against three systems built in the same harness. Two of them, autoregressive wait-k and chunkwise mask-predict, sit at the two ends of the parallelism spectrum and the third is a streaming analogue of recent block diffusion (Arriola et al., 2025), which is the closest non-autoregressive alternative for a competitive comparison. Another streaming translation system NAST (Ma et al., 2023) is also relevant, but its READ/WRITE policy is latent, emerging from alignment training, and its native setting is text-to-text where the policy conditions on content. ViNAYA makes the opposite trade and does not adapt to source content, which is what gives it the structural guarantee in Section 3.5.

Across all three corpora, ViNAYA cuts Average Lagging by 45 to 50 percent against the block-diffusion analogue, and it leads on both lag metrics on every corpus against the autoregressive baseline. Against block diffusion, quality is at or above parity in 10 of 12 cells. Autoregression retains a small quality edge on the two source-heavy corpora. On ActivityNet, where captions are longest, ViNAYA leads on every quality metric, on both lag metrics, and on wall-clock time.

Contributions

- We introduce yoked schedules: a text frontier and a source horizon running on one clock, both fixed in closed form before decoding begins (Section 3). To our knowledge this is the first streaming non-autoregressive captioner of any kind, and the first with a declared rather than learned schedule.
- We recover a flexible exposure budget. The expected fraction of source visible per emitted token, $\bar{E}$, is the quantity a streaming schedule is built to set. A single exponent pins it at $1/2$ for every value, while two exponents make it $1/(1+\nu)$ as a function of their ratio alone (Theorem 1).
- We provide guarantees that hold structurally, independent of training. Every emitted segment is final the moment it leaves, and logits depend strictly on the bounded source-feature prefix (Section 3.5 and Appendix A).
- We execute a controlled comparison of three streaming baselines for video captioning across three corpora that span both source-heavy and text-heavy regimes. Against the strongest non-autoregressive alternative, ViNAYA holds quality at parity while halving Average Lagging (Section 4).

2 Related Work

ViNAYA sits at the intersection of three lines of work: parallel sequence generation, streaming policies for simultaneous translation, and video captioning. We position against each in turn.

Parallel and block-causal sequence generation accelerates inference by breaking the traditional token-by-token bottleneck. Foundational non-autoregressive models, iterative refinement strategies, and discrete-diffusion frameworks successfully trade sequential rigidity for the ability to refine entire sequences in parallel (Gu et al., 2018; Lee et al., 2018; Ghazvininejad et al., 2019; Austin et al., 2021; Li et al., 2022). However, this acceleration fundamentally relies on dynamic, content-driven

unmasking across the full canvas. Because the write order is an emergent property of the model's confidence rather than a predetermined rule, these decoders have no monotone commitment, so no nontrivial prefix can be certified before decoding terminates. Recent block-diffusion language models (Arriola et al., 2025) partially address this limitation by committing text chunks sequentially, making them the closest architectural ancestor to our write schedule. Yet, designed for text-to-text generation, they have no mechanism to schedule the ingestion of an arriving source.

Streaming non-autoregressive generation has seen isolated developments, most notably within simultaneous machine translation. Systems such as NAST (Ma et al., 2023) achieve streaming capabilities by pairing a unidirectional encoder with a parallel decoder. To manage the flow of information, NAST relies on a latent, learned alignment policy that regulates token emission, encouraged by a latency-penalizing loss during training. While effective, this methodology treats the read/write schedule as a soft behaviour dependent on source content. ViNAYA takes the inverse approach: by defining the dual schedule mathematically before the first word is ever drafted, we replace loss-driven tendencies with strict, structural finality that safely guarantees monotonic generation regardless of the underlying trained weights.

Read/write policies set the balance between accumulated source context and emission latency. Streaming systems use either rigid prefix rules, such as wait-k (Ma et al., 2019; Elbayad et al., 2020), or dynamic attention thresholds (Chiu & Raffel, 2018; Ma et al., 2020). We adopt this operational framing alongside its standard latency metrics, Average Lagging (Ma et al., 2019) and its length-adaptive repair LAAL (Papi et al., 2022). In the autoregressive setting, recent work introduced a continuous single-exponent source horizon that reduces context dilution (Mehmood & Javed, 2026). ViNAYA adopts this schedule for the source side and extends it to the write side. Coupling the two frontiers through a ratio ν with a closed-form rule for choosing it from corpus geometry turns the write order into a declared parallel schedule rather than a sequential loop.

Streaming video captioning is our primary evaluation domain, and the streaming systems in it are autoregressive (Zhou et al., 2024). The existing parallel captioners in this modality are non-streaming: diffusion-based decoders (Wang et al., 2026), object-oriented decoders (Liu et al., 2021), and coarse-to-fine decoders (Yang et al., 2021) all generate from a complete video rather than from a live feed. To our knowledge, no streaming non-autoregressive standard exists in this modality, which is why we build the streaming block-diffusion baseline for the comparison. ViNAYA is the first to bring yoked parallel scheduling to visual streams.

3 Methodology: Yoked Schedules for Parallel Streaming

In this section we develop the yoked schedule that makes parallel decoding streamable. We begin by stating the obstruction that a standard parallel decoder faces, then introduce the two frontiers, prove the exposure budget, and close with the streaming guarantees.

3.1 Setup and the Non-Autoregressive Streaming Problem

A *window* is one annotated segment carrying F_c encoded source tokens $X = (x_0, \ldots, x_{F_c-1})$ from a frozen visual encoder and a caption of N tokens including EOS. Because video encoders process temporal chunks, our causality guarantee operates strictly at the feature-sequence level rather than on raw video frames. A *history* is the single preceding segment, right-aligned into a canvas of length $2M$, with the current caption occupying positions M onward so absolute positional codes remain invariant. We denote the source-to-target geometry as $\rho = F_c/N$.

Standard parallel decoders buy speed by generating many tokens per pass, but this speed comes at a structural cost we now make precise. In mask-predict architectures (Ghazvininejad et al., 2019), the unmasking order is determined at inference by model confidence rather than by a fixed rule, so no prefix can be certified before decoding terminates. This creates a structural obstruction to streaming, which we formalize as follows (proof in Appendix A):

Proposition 1 (Mask-predict certifies no prefix)**.** *Call an emitter* sound *if every symbol it releases equals the symbol at that position in the final output. For mask-predict with confidence-based unmasking and at least two rounds, no sound emitter releases a non-empty set of positions before the final round.*

Algorithm 1 ViNAYA training (one example)

Require: source X, target Y, history $(X_{\text{prev}}, Y_{\text{prev}})$

$N \leftarrow \max(|Y|, \hat{N}) + \delta,\ \delta \sim U\{-1, 0, 1, 2\}$
$n_0, \ldots, n_T \leftarrow$ text frontier from Eq. 1, length N
draw $j \sim U[0, \min(N, |Y|))$; $k \leftarrow$ span containing j
$\Omega_k \leftarrow \min(\max(\lceil F_c(k/T)^{\gamma_v} \rceil, \phi), F_c)$
draw inner round $s \sim U\{1, \ldots, S\}$
$n_{\text{mask}} \leftarrow |S_k|$ if s=1, else $\lfloor \cos(\frac{\pi}{2} \frac{s-1}{S}) |S_k| \rfloor$
canvas $\leftarrow$ history and target, all tokens clean
mask n_{mask} positions drawn uniformly from S_k
mask every position in $[n_k, N)$; set positions outside $[0, N)$ to PAD
return $\sum_{i \in S_k} w_i \, \text{CE}(\text{logits}_i, Y_i)$

Algorithm 2 ViNAYA streaming inference

Require: source X, history $(X_{\text{prev}}, Y_{\text{prev}})$

$\hat{N} \leftarrow \min(\max(Q_q(N \mid |Y_{\text{prev}}|) + \text{margin}, 3), M)$
$n_0, \ldots, n_T \leftarrow$ text frontier from Eq. 1, length $\hat{N}$
canvas $\leftarrow$ [SOS] $\cdot\, Y_{\text{prev}} \cdot$ [SEP], right-aligned into $[0, M)$
canvas$[M{:}M{+}\hat{N}] \leftarrow$ [MASK]; canvas$[M{+}\hat{N}{:}2M] \leftarrow$ [PAD]
for $k = 1$ **to** T **do**
  if $|S_k| = 0$ **then continue**
  end if
  $\Omega_k \leftarrow \min(\max(\lceil F_c(k/T)^{\gamma_v} \rceil, \phi), F_c)$
  for $s = 1$ **to** S **do**
    decode; unmask top-confidence positions in S_k
  end for
  $e_k \leftarrow$ canvas$[S_k]$ truncated at first EOS or PAD
  emit e_k; **if** e_k truncated **break**
end for

Because no position is excluded a priori from being re-predicted in a later round, a standard parallel decoder cannot certify any prefix as final until decoding terminates. ViNAYA repairs this by removing the dynamic unmasking order and replacing it with a declared schedule.

3.2 Two Frontiers on One Clock

Generation in ViNAYA runs on a unified clock $r_k = k/T$ for pass $k = 1, \ldots, T$. Two power laws are yoked to this single clock:

$$n_k = \min\Big(\hat{N},\ \max\left(n_{k-1},\ \lceil \hat{N}\, r_k^{\gamma_y} \rceil,\ k\right)\Big), \qquad \Omega_k = \min\Big(F_c,\ \max\left(\phi,\ \lceil F_c\, r_k^{\gamma_v} \rceil\right)\Big). \quad (1)$$

Here, n_k dictates the text frontier, establishing that the span written at pass k is $S_k = [\, n_{k-1}, n_k)$. The outer $\min$ caps the frontier at the predicted upper bound $\hat{N}$, while the inner $\max$ enforces monotonicity and bounds the global progress. Simultaneously, Ω_k dictates the source horizon, restricting cross-attention to the first Ω_k tokens of the current segment (plus the fully arrived history). A small floor ϕ guarantees the opening span has sufficient visual evidence.

At pass k, the model fills S_k, emits it, and freezes it. Everything below n_{k-1} is committed and never revised. The training loop is given in Algorithm 1 and the streaming inference loop is given in Algorithm 2.

3.3 The Exposure Budget and the Necessity of Two Exponents

The primary objective of a streaming schedule is to manage the exposure budget, the mean fraction of the clip visible when a token is committed. A natural first design might use a single exponent for both frontiers. However, doing so destroys the budget entirely.

Theorem 1 (The exposure budget)**.** *Let $\bar{E}$ denote the mean fraction of the clip visible at the moment a token is committed, averaged over token positions, in the continuum limit $N \to \infty$.*

(i) If a single exponent drives both frontiers ($\gamma_y = \gamma_v$), then in the continuum limit $\bar{E} = \frac{1}{2}$ for every exponent value.

(ii) With separate exponents, $\bar{E} = 1/(1{+}\nu)$ where $\nu = \gamma_v/\gamma_y$. It depends on the two exponents only through their ratio.

(iii) The realised schedule is a step function, and the quantity a finite-T run actually achieves is $\bar{E}_T = \sum_{k=1}^{T} \big(r_k^{\gamma_y} - r_{k-1}^{\gamma_y}\big) r_k^{\gamma_v}$, which converges to $1/(1+\nu)$ as $T \to \infty$.

Proof sketch. A token at fractional position $x = i/N$ is committed when the text frontier reaches it at $r = x^{1/\gamma_y}$. At this instant, the visible source fraction is $r^{\gamma_v} = x^{\gamma_v/\gamma_y} = x^{\nu}$. Averaging this exposure over all positions yields a Riemann sum for $\int_0^1 x^{\nu} dx = 1/(1+\nu)$. If $\nu = 1$ (a single exponent), the exponent cancels out, pinning the budget at $1/2$. The full proof, including the finite-T computation, is in Appendix A. □

Splitting the exponents recovers control over the exposure budget. By coupling them via the ratio $\nu = \gamma_v/\gamma_y$, the budget is fixed and the text exponent γ_y alone governs span allocation. Appendix B states the single-exponent case in full and derives the closed-form rule for choosing ν from corpus geometry.

3.4 Span Allocation and Sequential Cost

The ratio ν sets how much source each word costs. The text exponent γ_y then controls how the T passes are spread along the sentence. Differentiating the frontier gives an expected span size of $|S_k| \approx (N\gamma_y/T)\, r_k^{\gamma_y - 1}$, so $\gamma_y = 1$ yields equal blocks and $\gamma_y > 1$ places shorter spans at the opening, where the model has the least evidence, and wider spans at the close, where context is abundant.

Crucially, declaring this frontier in advance bounds the model's sequential cost independent of sequence length.

Proposition 2 (Bounded sequential cost)**.** *ViNAYA executes at most TS forward passes (where S is the number of refinement rounds) and emits at most T spans per window, independent of $\hat{N}$. A block decoder of block size L executes $\lceil \hat{N}/L \rceil S$ passes. The two coincide at $L = \lceil \hat{N}/T \rceil$.*

3.5 Structural Streaming Guarantees

Because the frontiers n_k and Ω_k are fixed prior to decoding, ViNAYA provides strict streaming guarantees that are structural rather than learned. They hold for randomly initialised weights exactly as they do for trained ones. Complete proofs are in Appendix A.

The first guarantee adapts the dependency theorem of Mehmood & Javed (2026) to the parallel setting. The argument is the same induction over layers, but the readable canvas set is now a span-bounded prefix rather than an autoregressive one.

Theorem 2 (Dependency)**.** *Under the yoked schedule, for any query position i in span S_k, the logits produced at pass k depend only on the canvas prefix below n_{k-1}, the current span's own partial contents, and the source-feature prefix $X_{<\Omega_k}$. They depend on no source feature token at or beyond Ω_k.*

While Theorem 2 governs what the model is permitted to read, the second guarantee establishes what it can safely release. In an autoregressive decoder, incremental emission is trivial: writing one token per step naturally yields a final prefix. In ViNAYA, several positions are committed at once from a partial canvas, so prefix finality has to be established rather than assumed.

Theorem 3 (Exact incremental emission)**.** *Every emitted span is final at the instant it is emitted. The sequence delivered to the consumer after any number of passes is a prefix of the final output.*

Theorem 3 directly resolves the obstruction of Proposition 1.

3.6 Implementation: Length Estimation and Training

To set the schedule without peeking at the unarrived source, $\hat{N}$ is estimated exclusively from the previous window's text: $\hat{N} = \min\big(\max(Q_q(N \mid |Y_{\text{prev}}|) + \text{margin},\ 3),\ M\big)$, where Q_q is the q-th quantile of the smoothed conditional length distribution. Algorithm 1 gives the exact training procedure.

During training, the canvas length is jittered ($N = \max(N_{\text{true}}, \hat{N}) + \delta, \delta \sim U\{-1, 0, 1, 2\}$) so that EOS is not always in the last slot, forcing the model to place it by semantics rather than position. Because $\gamma_y \neq 1$ results in spans of varying sizes, queries lack implicit information about span boundaries. We repair this by injecting a block ID and the exposure state $(\Omega_k/F_c,\ k/T)$ directly into the token embeddings. Appendix C gives the run configuration.

4 EXPERIMENTS

Our experiments establish three findings. ViNAYA reduces Average Lagging significantly against the strongest streaming non-autoregressive baseline, at quality parity in most cells. The closed-form exposure budget predicts the realised exposure on three corpora that span both source-heavy and text-heavy regimes. And at a fixed budget, moving only the text exponent changes quality by up to 12 percent on BLEU-4, which shows that the two dials act on different aspects of the schedule rather than duplicating each other.

4.1 DATASETS

We evaluate on three video captioning corpora whose source-to-target geometry spans both regimes. YouCook2 (Zhou et al., 2018) and Charades-STA (Gao et al., 2017) are source-heavy, with $\rho = 2.44$ and $\rho = 2.62$ respectively: a clip of roughly twenty encoder tokens funds a caption of eight or nine subword tokens, so the source must lead. ActivityNet Captions (Krishna et al., 2017) has $\rho = 0.71$ and is text-heavy, with a sparser visual source supporting a longer caption. ActivityNet's geometry gives the schedule the least room to help, and we report it as the corpus where we expect the smallest gains.

Three frozen encoders are used, one per corpus, so that any measured difference can be attributed to the schedule rather than to representation quality. YouCook2 uses the AVLnet features (Rouditchenko et al., 2021): a ResNet-152 2D stream at 1 fps concatenated with a 3D ResNeXt-101 stream at 24 fps in 16-frame chunks. Charades-STA uses CLIP ViT-B/32 (Radford et al., 2021) at 3 fps. ActivityNet uses per-segment C3D fc7 slices, cut from the whole-video feature array using the official timestamps. Data preprocessing details, including how streaming pairs are formed, are provided in Appendix D.

All systems freeze the visual encoder and train the decoder from scratch. Every system in the comparison shares the same trunk: 4 layers, d=768, 8 heads, feed-forward 1536, dropout 0.1, and a per-corpus BPE vocabulary. All systems are trained with AdamW (Loshchilov & Hutter, 2019) under cosine decay with warmup, batch size 8, and early stopping on validation loss.

4.2 SYSTEMS AND PROTOCOL

All baselines are the same program with one integer changed. The autoregressive wait-k baseline (L=1) writes one token per block and takes one pass per token. The block-diffusion analogue (L=4) uses fixed blocks with block-causal attention and S rounds per block, which is the strongest streaming analogue of block diffusion (Arriola et al., 2025) in this setting. Because no off-the-shelf non-autoregressive streaming baseline exists for video captioning, we constructed an architecturally matched and identically trained baseline within our own harness, sharing the trunk, canvas, length prior, and learning schedule with ViNAYA to ensure a strictly controlled comparison. Chunkwise mask-predict ($L \geq \hat{N}$) collapses to one block per window. Every system shares the trunk, canvas, counted length prior, loss vector, and optimiser by construction.

ViNAYA uses T=4 passes, γ_y=1.5, ν=0.5 (hence γ_v=0.75, β_1=0.354), and S=2 inner rounds per pass. The source floor ϕ is 5 on the two source-heavy corpora and 3 on ActivityNet, whose median clip is short enough that a floor of 5 would dominate the first two horizons. All three baselines share the trunk, canvas, counted length prior, loss vector, and optimiser with ViNAYA by construction. The block-diffusion baseline is the closest streaming non-autoregressive alternative. The autoregressive baseline provides a reference point for the quality ceiling that NAR systems are conventionally measured against. All systems were trained and evaluated on a single T4 GPU per experiment on Kaggle or Google Colab. Preprocessing, training, and inference scripts, together with the instructions to use are provided in the supplementary material.

We report BLEU-1 and BLEU-4 (B-1, B-4) (Papineni et al., 2002), CIDEr (Vedantam et al., 2015), METEOR (MET) (Banerjee & Lavie, 2005), Average Lagging (AL) (Ma et al., 2019) and its length-adaptive repair LAAL (Papi et al., 2022), with both lengths in subword tokens, measured forward passes per window, and measured wall-clock inference time. ROUGE-L (R-L) (Lin, 2004) is additionally reported in Tables 4 and 5.

4.3 Main results

Table 1 reports the full streaming comparison. Every ViNAYA row uses the headline configuration: T=4, γ_y=1.5, ν=0.5 (hence γ_v=0.75), S=2 inner rounds, and a per-corpus source floor ϕ. Every baseline runs on the same trunk, canvas, counted length prior, and optimiser, with only the block schedule or source-horizon rule changed. Each cell is averaged over three seeds.

Corpus	System	B-1	B-4	CIDEr	MET	AL	LAAL	passes
YouCook2	AR wait-k	0.2502	0.0428	**0.4997**	**0.2128**	4.24	7.54	7.27
YouCook2	BD3-LM	0.2617	0.0385	0.4660	0.2092	7.15	10.23	**4.21**
YouCook2	Chunkwise	0.1637	0.0078	0.1169	0.1377	29.92	29.92	2.00
YouCook2	**ViNAYA**	**0.2674**	**0.0419**	0.4629	0.2054	**3.58**	**6.25**	4.52
Charades	AR wait-k	**0.3111**	**0.0614**	**0.5434**	**0.2493**	4.79	6.41	7.15
Charades	BD3-LM	0.3057	0.0508	0.4541	0.2349	7.98	9.53	**4.31**
Charades	Chunkwise	0.2596	0.0216	0.1446	0.2055	23.44	23.44	2.00
Charades	ViNAYA	0.3070	0.0539	0.4642	0.2368	**4.13**	**5.83**	5.96
ActivityNet	AR wait-k	0.1581	0.0178	0.1162	0.1153	3.39	4.99	14.09
ActivityNet	BD3-LM	0.1826	0.0202	0.1666	0.1282	3.89	5.55	6.95
ActivityNet	Chunkwise	0.1196	0.0061	0.0408	0.0976	16.76	16.76	2.00
ActivityNet	**ViNAYA**	**0.1950**	**0.0228**	**0.1806**	**0.1332**	**2.15**	**3.69**	**4.70**

Table 1: Streaming results, 3 seeds per system.

System	Total (s)	ms/window	Relative
Chunkwise	95.7	53.7	0.53×
ViNAYA	**179.3**	**100.6**	1.00×
BD3-LM	229.7	128.9	1.28×
AR wait-k	426.7	239.4	2.38×

Table 2: Wall-clock inference on the ActivityNet test split (1,782 windows, three seeds). ViNAYA is 28 percent faster than the strongest streaming non-autoregressive baseline and 2.38 times faster than autoregression on the corpus with the longest captions.

ViNAYA cuts Average Lagging by 50%, 48%, and 45% and LAAL by 39%, 39%, and 33% across the three corpora, leading the autoregressive reference on both lag metrics on every corpus. Quality is at or above parity with the block-diffusion baseline in 10 of 12 cells, the two exceptions being CIDEr and METEOR on YouCook2. On the two metrics the bootstrap reports, the losses on YouCook2 are statistically significant (Section 4.7). Autoregression retains a small quality edge on the two source-heavy corpora, but on ActivityNet, where captions are longest, ViNAYA leads on every quality metric, on both lag metrics, and on wall-clock time. The pass-count column confirms the crossover that Proposition 2 predicts: ViNAYA holds a constant $T \cdot S = 8$ passes per window, so it costs 7 to 38 percent more than the block baseline on the short-caption corpora and 32 percent less on ActivityNet. Wall-clock inference on that corpus (Table 2) is 179 seconds, 28 percent below block diffusion and 2.38 times faster than autoregression.

4.4 Theory validation: the exposure budget

Table 3 tests the budget prediction of Theorem 1 by sweeping ν at fixed γ_y=1.5 and T=4. The continuum limit $\bar{E} = 1/(1+\nu)$ and the finite-T closed form $\bar{E}_T$ are drawn from the schedule alone. The measured exposure $\bar{E}$ meas is the fraction of source visible per emitted token, averaged over the decodes. The three values agree as the theory requires, and the residual gap between $\bar{E}_T$ and

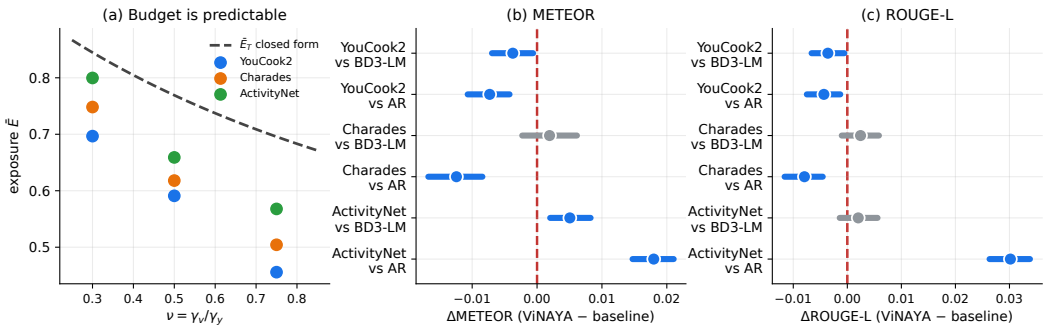


Figure 1: (a) Measured exposure $\bar{E}$ against ν with the closed-form curve $\bar{E}_T$ (dashed). (b, c) Paired bootstrap confidence intervals for ViNAYA-to-baseline deltas on METEOR and ROUGE-L. Blue intervals exclude zero at 95%; grey straddle it.

measurement is smallest on ActivityNet, where the source floor ϕ binds on short clips and offsets the truncation bias that dominates on the longer corpora. Figure 1(a) plots the same sweep, with the closed-form curve drawn in advance from the schedule and the three corpora overlaid.

Corpus	ν	$\bar{E}$	$\bar{E}_T$	$\bar{E}$ meas	gap
YouCook2	0.30	0.769	0.845	0.6967	0.148
YouCook2	0.50	0.667	0.769	0.5910	0.178
YouCook2	0.75	0.571	0.696	0.4558	0.240
Charades	0.30	0.769	0.845	0.7482	0.097
Charades	0.50	0.667	0.769	0.6180	0.151
Charades	0.75	0.571	0.696	0.5043	0.192
ActivityNet	0.30	0.769	0.845	0.7997	0.045
ActivityNet	0.50	0.667	0.769	0.6590	0.110
ActivityNet	0.75	0.571	0.696	0.5678	0.128

Table 3: Exposure sweep at fixed γ_y=1.5. $\bar{E}$ is the continuum limit, $\bar{E}_T$ the finite-T closed form, and $\bar{E}$ meas the value measured from the decodes. The gap column is $\bar{E}_T$ minus the measured value.

4.5 Shape at fixed exposure

Table 4 isolates span shape by holding ν=0.5 fixed, so $\bar{E}$ is pinned at 0.667 across all rows and only γ_y moves. Each cell is averaged over three seeds. Under this fixed budget, γ_y alone moves BLEU-4 by up to 12 percent on Charades, 10 percent on YouCook2, and 42 percent on ActivityNet. A schedule spending short spans early and widening as context accumulates is measurably better than one that spreads the passes evenly. γ_y=1.5 is balanced across the corpora; on ActivityNet, γ_y=2.5 trades CIDEr and ROUGE-L for a BLEU gain.

4.6 Offline ceiling

Table 5 contrasts each streaming model against the same weights run with Ω_k=F_c at every pass. All rows are three-seed averages. On the corpora with shorter captions (YouCook2 and Charades), an oracle decoder waiting for the full clip establishes a clear quality ceiling, improving metrics at the cost of roughly six to eight times the lagging. On ActivityNet, where captions are longest, this full-context advantage vanishes, and the streaming configuration matches or slightly leads the oracle on most metrics. This suggests that for longer sequences, restricting the source horizon focuses attention and improves generation rather than acting strictly as an informational cost.

Corpus	γ_y	B-1	B-4	CIDEr	R-L	MET	AL
Charades	1.0	**0.3106**	0.0475	0.4239	0.3271	0.2307	4.34
Charades	1.5	0.3070	**0.0539**	0.4642	0.3297	**0.2368**	4.13
Charades	2.5	0.2983	0.0506	**0.4713**	**0.3325**	0.2357	**3.54**
YouCook2	1.0	0.2730	0.0376	0.4482	**0.2866**	0.2046	4.76
YouCook2	1.5	0.2674	**0.0419**	**0.4629**	0.2830	**0.2054**	**3.58**
YouCook2	2.5	**0.2870**	0.0346	0.3844	0.2741	0.1976	4.02
ActivityNet	1.0	0.1832	0.0167	0.1609	0.1876	0.1259	2.51
ActivityNet	1.5	0.1950	0.0228	**0.1806**	**0.1895**	**0.1332**	**2.15**
ActivityNet	2.5	**0.2092**	**0.0237**	0.1558	0.1854	0.1309	2.52

Table 4: Span shape at fixed exposure, ν=0.5 across all rows.

Corpus	System	B-1	B-4	CIDEr	R-L	MET	AL
YouCook2	streaming	0.2674	0.0419	0.4629	0.2830	0.2054	**3.58**
YouCook2	offline	**0.2806**	**0.0434**	**0.4820**	**0.2896**	**0.2142**	29.92
Charades	streaming	0.3070	0.0539	0.4642	0.3297	0.2368	**4.13**
Charades	offline	**0.3281**	**0.0650**	**0.5242**	**0.3420**	**0.2560**	23.44
ActivityNet	streaming	**0.1950**	**0.0228**	0.1806	**0.1895**	**0.1332**	**2.15**
ActivityNet	offline	0.1901	0.0227	**0.1873**	0.1885	0.1316	16.76

Table 5: Offline ceiling versus streaming at γ_y=1.5.

4.7 Statistical significance

Figure 1(b) and (c) report paired bootstrap confidence intervals for the ViNAYA-to-baseline deltas on METEOR and ROUGE-L, over 1000 window-level resamples pooled across seeds. Against the block-diffusion baseline, ViNAYA trails on both metrics on YouCook2, holds parity on Charades, and registers a significant win on METEOR on ActivityNet. Against the autoregressive reference, ViNAYA trails on YouCook2 and Charades and leads on both metrics on ActivityNet. The ordering reverses on the corpus with the longest captions, which is what the geometry predicts.

5 Summary, Limitations and Future Work

ViNAYA makes streaming non-autoregressive captioning possible by treating the read and write order as a schedule to be declared rather than learned. Two frontiers advance on one clock: a monotone source horizon that reveals the clip, and a monotone text frontier that commits and emits spans from left to right. No streaming non-autoregressive standard exists in this modality, and the streaming block-diffusion baseline we build for the comparison (Arriola et al., 2025) is the closest non-autoregressive alternative. Against that baseline, ViNAYA reduces Average Lagging by 45 to 50 percent across three corpora with three frozen encoders, at quality at or above parity in ten of twelve cells. Against an autoregressive reference, it leads on both latency metrics on every corpus, and on ActivityNet, where captions are longest, it leads on quality, latency, and wall-clock time.

On the theoretical side, Theorem 1 gives a closed-form expected budget that we validate against measurement on all three corpora, and the ratio ν binds the source reveal to the text schedule so that only γ_y needs to be swept. Theorem 2 and Theorem 3 are the load-bearing structural guarantees and hold independent of training.

While the exposure ratio ν is grounded analytically in corpus geometry, the text exponent γ_y currently operates as a fixed corpus-level parameter. In future work, we plan to eliminate this sweep by learning the text schedule jointly during training, allowing the frontier to adapt dynamically to source content. Additionally, we plan to extend ViNAYA to a full discrete-diffusion formulation that reduces the sequential cost further, and to evaluate it on more complex streaming tasks such as online sign language translation (SLT).

AI USE STATEMENT

We acknowledge the use of generative AI tools for manuscript polishing (specifically for grammar correction and condensing text), generating boilerplate code, and complementing our literature search. All AI usage was judicious and underwent rigorous human verification. The primary research ideas, hypotheses, identification of gaps, research methodology, and theoretical contributions belong exclusively to the authors. We take full responsibility for the final content of this work, including all text, claims, and artifacts. Below is the detailed disclosure of AI usage across all *required* research tasks:

Research Task	**AI Used**	**Details & Verification**
Generate synthetic data sets	No	
Develop theoretical models or conceptual frameworks	No	
Formulate mathematical claims	No	
Provide critical ingredients for proving mathematical claims	No	
Assist in the writing of proofs	Partial	Formatted proofs for clean LaTeX presentation; all proof arguments are the authors' own
Propose or refine hypotheses	No	
Design or provide feedback on research methodology or experiments	No	
Implement methods	Partial	Generated boilerplate code (e.g., standard neural network architectures) & polishing before release; adopted after verification
Assist with translation	N/A	
Clean and reformat dataset	No	
Support qualitative and thematic data analysis	No	
Interpret results	No	

REPRODUCIBILITY STATEMENT

All training and inference code, corpus preparation scripts, feature extraction pipelines, and the execution commands required to reproduce the reported results in Tables 1–5 are included in the supplementary material. The two proof appendices (A and B) cover every formal claim.

ETHICS STATEMENT

This work uses three publicly released video captioning corpora under their respective licences, and pre-extracted frozen features distributed by their original authors. No new data was collected, no human subjects were involved, and no personally identifying information is used or released.

REFERENCES

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, 2005.

Chung-Cheng Chiu and Colin Raffel. Monotonic chunkwise attention. In *International Conference on Learning Representations (ICLR)*, 2018.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. Efficient wait-k models for simultaneous machine translation. In *Interspeech*, pp. 1461–1465, 2020.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: Temporal activity localization via language query. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 5267–5275, 2017.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6112–6121, 2019.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. Non-autoregressive neural machine translation. In *International Conference on Learning Representations (ICLR)*, 2018.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 706–715, 2017.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1173–1182, 2018.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, 2004.

Fenglin Liu, Xuancheng Ren, Xian Wu, Bang Yang, Shen Ge, and Xu Sun. O2NA: An object-oriented non-autoregressive approach for controllable video captioning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 281–292, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3025–3036, 2019.

Xutai Ma, Juan Pino, James Cross, Liezl Puzon, and Jiatao Gu. Monotonic multihead attention. In *International Conference on Learning Representations (ICLR)*, 2020.

Zhengrui Ma, Shaolei Zhang, Shoutao Guo, Chenze Shao, Min Zhang, and Yang Feng. Non-autoregressive streaming transformer for simultaneous translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5177–5190, 2023.

Yasir Mehmood and Kashif Javed. Reading less while writing: A closed-form bandwidth dial for streaming multimodal decoders, 2026. URL `https://arxiv.org/abs/2609.20845`.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. Over-generation cannot be rewarded: Length-adaptive average lagging for simultaneous speech translation. In *Proceedings of the Third Workshop on Automatic Simultaneous Translation*, pp. 12–17, 2022.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318, 2002.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Andrew Rouditchenko, Angie Boggust, David Harwath, Brian Chen, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Hilde Kuehne, Rameswar Panda, Rogerio Feris, Brian Kingsbury, Michael Picheny, Antonio Torralba, and James Glass. AVLnet: Learning audio-visual language representations from instructional videos. In *Interspeech*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Junbo Wang, Liangyu Fu, Yuke Li, Yining Zhu, Ya Jing, Xuecheng Wu, and Jiangbin Zheng. DiffVC: A non-autoregressive framework based on diffusion model for video captioning. *arXiv preprint arXiv:2604.08084*, 2026.

Bang Yang, Yuexian Zou, Fenglin Liu, and Can Zhang. Non-autoregressive coarse-to-fine video captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, 2018.

Xingyi Zhou, Anurag Arnab, Shyamal Buch, Shen Yan, Austin Myers, Xuehan Xiong, Arsha Nagrani, and Cordelia Schmid. Streaming dense video captioning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

A Full Proofs

A.1 Notation and standing assumptions

Canvas positions are $0, \dots, 2M-1$. Let $\mathrm{canvas}[j]$ denote the token at canvas position j and $\mathrm{blk}[j]$ its span assignment: $\mathrm{blk}[j] = -1$ for the history block, $\mathrm{blk}[j] = k$ for positions in span S_k, $\mathrm{blk}[j] = T+1$ for positions beyond the text frontier, and positions outside $[0, \hat{N})$ carry the padding symbol. Define the readable canvas set

$$V_b = \{\, j : \mathrm{blk}[j] \le b,\ \mathrm{canvas}[j] \neq \textsc{pad} \,\}$$

and the readable source set

$$U_k = [0, F_p) \ \cup\ [F_{\max},\ F_{\max} + \Omega_k),$$

where $F_{\max}$ is the padded per-window source width and F_p the true length of the previous window.

For a set S of positions, $\mathrm{canvas}|_S$ denotes the restriction of the canvas to S, and for an index range R, $X|_R$ denotes the restriction of the source to R. Let $y^{(k,s)}$ denote the canvas after inner round s of pass k, and $P(S_k)$ the canvas positions of span S_k.

Assumption 1. *The predicted length $\hat{N}$, the text frontier $n_0, \dots, n_T$ and the source horizons $\Omega_1, \dots, \Omega_T$ are fixed before decoding begins. Masked attention logits are set to $-\infty$ exactly. The decoder is built exclusively from self-attention, cross-attention, and position-wise layers.*

Lemma 1 (Monotone frontiers and write locality)**.** *(i) The text frontier satisfies $0 = n_0 \le n_1 \le \cdots \le n_T = \hat{N}$, and $k \mapsto \Omega_k$ is non-decreasing and bounded by F_c. (ii) During pass k, the decoder modifies only positions in $P(S_k)$. Thus, for every $j \in P(S_1 \cup \cdots \cup S_k)$, every $k' > k$, and every inner round s, $y_j^{(k',s)} = y_j^{(k,S)}$.*

Proof. (i) In Eq. 1, the inner $\max$ takes the running maximum with n_{k-1} and the outer $\min$ caps at $\hat{N}$, so both operations preserve monotonicity. Since $r \mapsto r^{\gamma_v}$ is non-decreasing on $[0,1]$ for $\gamma_v \geq 0$, the composition is non-decreasing. (ii) The unmasking step is restricted to positions still masked *in* S_k. Because the spans are disjoint by (i), positions in prior spans lie outside $P(S_{k'})$ and are never reassigned. □

A.2 Proof of Proposition 1

Proof. Fix the number of rounds $R \geq 2$. At round $r < R$, mask-predict ranks *all* positions by model confidence under the current canvas; no position is excluded a priori from being masked in a later round. Let $\mathcal{E}$ be an emitter that releases a non-empty position set P after round $r < R$. It suffices to exhibit a decoder whose logits alternate between two outputs depending on whether some position beyond $\max P$ is masked. Take parameters that realise this predicate: the network outputs symbol a at every position whenever the canvas contains a mask symbol at an index greater than $\max P$, and symbol $b \neq a$ once no such mask remains.

The round-R canvas differs from the round-r canvas exactly in this respect whenever some position beyond $\max P$ is still masked at round r. The value released at P is a, but the final value is b, making $\mathcal{E}$ unsound. □

A.3 Proof of Theorem 1

Proof. **Parts (i) and (ii).** A token at fractional position $x = i/N$ is committed where $r^{\gamma_y} = x$, which is at $r = x^{1/\gamma_y}$. The visible fraction is $r^{\gamma_v} = x^{\gamma_v/\gamma_y} = x^{\nu}$. Averaging over the N positions yields $\bar{E} = \frac{1}{N}\sum_{i=1}^{N}(i/N)^{\nu}$, a Riemann sum for x^{ν} on $[0,1]$. Because $\nu \geq 0$, the sum converges to $\int_0^1 x^{\nu} dx = 1/(1+\nu)$. If $\gamma_y = \gamma_v$, then $\nu = 1$ and $\bar{E} = 1/2$ for all $\gamma > 0$. Rounding perturbs each horizon by at most one source token, contributing $O(1/F_c)$ per step, which is absorbed if ρ is bounded below.

Part (iii). In finite-T execution, every token of span k is committed with Ω_k evaluated at the *end* of its clock interval r_k. The fraction of tokens in span k is $r_k^{\gamma_y} - r_{k-1}^{\gamma_y}$. The collective exposure is $(r_k^{\gamma_y} - r_{k-1}^{\gamma_y}) r_k^{\gamma_v}$. Summing gives $\bar{E}_T$. As $T \to \infty$ this Riemann–Stieltjes sum converges:

$$\int_0^1 r^{\gamma_v}\, d(r^{\gamma_y}) = \gamma_y \int_0^1 r^{\gamma_y+\gamma_v-1}\, dr = \frac{\gamma_y}{\gamma_y+\gamma_v} = \frac{1}{1+\nu}.$$

□

A.4 Proof of Proposition 2

Proof. ViNAYA executes an outer loop of T passes, each running at most S inner rounds and emitting at most one span, yielding at most TS passes and T emissions, independent of $\hat{N}$. A block decoder with block size L partitions $[0, \hat{N})$ into $\lceil \hat{N}/L \rceil$ blocks running S rounds per block, giving $\lceil \hat{N}/L \rceil S$ passes. Equating $\lceil \hat{N}/L \rceil = T$ gives $L = \lceil \hat{N}/T \rceil$ as the unique block size where the budgets agree. □

A.5 Proof of Theorem 2

Proof. By induction on layer depth ℓ.

Base case ($\ell = 0$): $h_i^{(0)}$ is derived from the canvas token, positional table, block embedding, and $\mathrm{expo}_k = (\Omega_k/F_c,\ k/T)$. These depend only on $\mathrm{canvas}|_{V_{\mathrm{blk}[i]}}$ and constants fixed prior to decoding.

Inductive step: Assume $h_j^{(\ell-1)}$ depends only on $\big(\mathrm{canvas}|_{V_{\mathrm{blk}[j]}},\ X|_{U_k}\big)$. For self-attention, masked logits are exactly $-\infty$, receiving zero softmax weight. If $\mathrm{blk}[j] \leq \mathrm{blk}[i]$, then $V_{\mathrm{blk}[j]} \subseteq V_{\mathrm{blk}[i]}$ due to nestedness. Thus, every valid $h_j^{(\ell-1)}$ depends only on $\mathrm{canvas}|_{V_{\mathrm{blk}[i]}}$ and $X|_{U_k}$. For cross-attention, $\beta_{im} = 0$ unless $m \in U_k$; the source mask is identical for every query at pass k, so the

output depends strictly on $X|_{U_k}$. Remaining maps are position-wise. Therefore, $h_i^{(\ell)}$ depends solely on $\big(\mathrm{canvas}|_{V_{\mathrm{blk}[i]}}, X|_{U_k}\big)$. Specialising to $\mathrm{blk}[i] = k$ yields the claim. □

A.6 Proof of Theorem 3

Proof. Let $c_k = \big(y_j^{(k,S)}\big)_{j \in P(S_k)}$ be the contents of span k at the end of its pass. By Lemma 1, $P(S_k)$ is written only during pass k, so c_k is fixed at the end of pass k and never changes. Let e_k be the truncated span removing tokens post-EOS/PAD. The emitter appends spans in increasing order, meaning $E_k = E_{k-1} \cdot e_k$. Right-concatenation is monotone for the prefix order, meaning $E_{k-1} \preceq E_k$. By induction, $E_k \preceq E_\infty$. Every symbol released occupies a position that is never rewritten, precisely matching the final output. □

B The Single-Exponent Schedule

The yoked schedule of Section 3 extends the single-exponent source horizon of a prior work (Mehmood & Javed (2026)) to two coupled frontiers. This appendix makes the transition explicit. Section B.1 restates the single-exponent schedule and its exposure budget. Section B.2 shows that a shared exponent across both frontiers collapses the budget to one half, which is Theorem 1(i) stated in the continuum. Section B.3 derives the closed-form rule for choosing the ratio ν from corpus geometry. Sections B.2 and B.3 are the paper's contribution; the first subsection is background.

B.1 The schedule and its budget

The single-exponent horizon and its budget were introduced by Mehmood & Javed (2026); we restate them here so the two-frontier construction below is self-contained. Let $X = (X_0, \ldots, X_{F-1})$ be F source tokens from a frozen encoder, $\hat{N}$ a predicted target length, and $i = 1, \ldots, \hat{N}$ the generation step. Define the horizon

$$\Omega_i = \big\lceil F \cdot (i/\hat{N})^\gamma \big\rceil, \qquad \gamma \geq 0.$$

At step i the decoder attends only to $X_{<\Omega_i}$. The mean source exposure per emitted token is

$$\bar{E}(\gamma) = \frac{1}{\hat{N}} \sum_{i=1}^{\hat{N}} \Big(\frac{i}{\hat{N}}\Big)^\gamma + O(1/F),$$

where the $O(1/F)$ term absorbs the ceiling. The sum is a Riemann sum for x^γ on $[0, 1]$, so $\bar{E}(\gamma) \to \int_0^1 x^\gamma \, dx = 1/(1+\gamma)$. Two properties follow. $\gamma = 0$ recovers full exposure. $\gamma = 1$ halves it. The rounding error is absorbed whenever the source-to-target ratio $\rho = F/\hat{N}$ is bounded below.

B.2 The collapse under a shared exponent

Suppose the same exponent γ from the single-frontier schedule drives both a text frontier and a source horizon in a parallel decoder, so $n_k = \lceil \hat{N} r_k^\gamma \rceil$ and $\Omega_k = \lceil F_c r_k^\gamma \rceil$ on the shared clock $r_k = k/T$. A token at fractional position $x = i/N$ is committed at the first pass whose frontier reaches it, that is at $r = x^{1/\gamma}$. At that instant the visible source fraction is $r^\gamma = x^{\gamma/\gamma} = x$, and the exponent has cancelled. Averaging over positions gives a Riemann sum for $\int_0^1 x \, dx = 1/2$. Every value of $\gamma > 0$ therefore pins $\bar{E}$ at one half, and the dial loses control over the quantity it was designed to set. This is Theorem 1(i) stated in the continuum.

B.3 Choosing the ratio from geometry

The two-exponent yoking replaces the shared γ with a ratio $\nu = \gamma_v/\gamma_y$. The budget becomes $\bar{E} = 1/(1+\nu)$ (Theorem 1(ii)), which depends on the two exponents only through their ratio. To fix ν from the corpus rather than by hand, we target a first-pass horizon of B source tokens: $\Omega_1 = \lceil F_c T^{-\nu\gamma_y} \rceil = B$. Solving for ν,

$$\nu^\star = \frac{\ln(F_c/B)}{\gamma_y \ln T}.$$

A larger tolerated buffer B gives a smaller $\nu^\star$ and a larger exposure budget. This is the two-frontier analogue of a geometry-derived exponent in the autoregressive setting, with the number of passes T taking the role of the generation horizon. We hold $\nu = 0.5$ in every headline experiment for simplicity and sweep it only in Section 4 to show its effect.

C Run Configurations

Table 6 details the complete hyperparameter, optimisation, and schedule configurations used for both ViNAYA and the baseline models.

	Setting	Value
Trunk	layers / width / heads / FFN	4 / 768 / 8 / 1536
	dropout	0.1
	positional encoding	fixed sinusoidal, absolute
	schedule conditioning	block embedding + dense($\Omega_k/F_c,\ k/T$)
	vocabulary	per-corpus BPE, 8k requested
	decoder-block parameters	28.4M (plus corpus-dependent embeddings)
Optimisation	optimiser	AdamW, weight decay 10^{-4}, clip-norm 1.0
	schedule	cosine decay, peak 10^{-4}, 2 warmup epochs
	batch size	8
	epochs	≤ 100; early stopping patience 7 (YouCook2, Charades) or 8 (ActivityNet) on val loss
	checkpoint selection	best deterministic validation loss
Schedule	T (ViNAYA) / L (baseline)	4 / 4 *(system)*
	γ_y, ν, γ_v	1.5, 0.5, 0.75
	$\beta_1 = T^{-\gamma_v}$	0.354
	source floor ϕ	5 (YouCook2, Charades); 3 (ActivityNet)
	inner rounds S	2
	unmasking rule	keep $\lfloor \cos(\frac{\pi}{2}\frac{s-1}{S})\lvert S_k\rvert \rfloor$ masked, fill the rest by confidence
Length prior	quantile q / margin	0.98 / 1
	smoothing	$\alpha = 5$ toward the marginal
	canvas jitter δ	$U\{-1, 0, 1, 2\}$ on $\max(N_{\text{true}}, \hat{N})$
Loss	draft-field weight	0 *(system: baseline has no such field)*

Table 6: Run configuration. Every entry is identical for ViNAYA and for the block baseline except the two marked "system".

D Preprocessing Details

YouCook2 features follow the AVLnet release (Rouditchenko et al., 2021), with official captions rejoined on (video id, segment index). Charades-STA uses CLIP ViT-B/32 (Radford et al., 2021) at 3fps with overlapping consecutive segments excluded when forming streaming pairs. ActivityNet Captions uses per-segment C3D slicing with the per-video sampling rate derived as feature length over duration.