Visual Text Compression as Measure Transport

Lv Tang[1*], **Tianyi Zheng**[3,*], **Yang Liu**[4], **Bo Li**[3,†], **Xingyu Li**[1,2†]
Department of Electrical and Computer Engineering, University of Alberta[1],
Alberta Machine Intelligence Institute (Amii)[2],
vivo BlueImage Lab, vivo Mobile Communication Co., Ltd.[3],
School of Vehicle and Mobility, Tsinghua University[4]
luckybird1994@gmail.com, libra@vivo.com, xingyu@ualberta.ca

Abstract

Visual text compression (VTC) promises efficient long-context processing by rendering text into an image and re-encoding it with a vision-language model, often producing 3–20$\times$ fewer decoder tokens than subword tokenization. Yet token savings do not translate predictably into downstream utility: on some tasks the visual path matches or exceeds the text path, on others it collapses, and the compression ratio itself does not predict which regime will occur. The missing quantity is therefore not another summary of efficiency, but a principled measure of task-relevant information loss induced by visual encoding. We address this problem by formulating VTC in the language of measure transport. Treating text and visual tokens as empirical probability measures, we show that the ViT patch encoder induces a push-forward map whose transport cost decomposes into a precision cost from within-patch aggregation and a coverage cost from cross-patch fragmentation. Both terms are estimable from downstream-label-free probes. This formulation yields two operational consequences: a downstream-label-free routing criterion that selects whether to use the visual path for a given input or benchmark instance, and a transport-informed foveation mechanism that re-encodes high-cost regions at higher resolution. Across 24 NLP datasets at Qwen3-4B, our label-free rule matches the per-dataset oracle on $17/24$ datasets (70.8%), and improves the average task score by $+3.3\%$ with -10.3% average tokens relative to a pure-LLM.

1 Introduction

Visual text compression (VTC) has recently emerged as a practical route to long-context processing. By rendering text into images and re-encoding it through a vision-language model (VLM), VTC replaces long subword-token sequences with substantially fewer visual tokens, often at compression ratios of 3–20$\times$ [9, 18, 23, 46–48]. DeepSeek-OCR [9] and Glyph [46] show that VLMs can preserve strong OCR fidelity and competitive long-context modeling even under heavy compression. These results establish the efficiency promise of VTC. They do not resolve the more fundamental question that determines whether VTC is actually usable: once text is rendered and re-encoded, what task-relevant information is lost, and when does that loss matter?

We answer this question by systematically evaluating VTC on a diverse suite of NLP benchmarks spanning understanding, reasoning, and generation. The outcome is sharply non-uniform (Tab. 1). On some tasks the visual path substantially outperforms the text path, while on others it falls far below. The amount of compression achieved has no predictive power over which outcome occurs. Datasets with the largest token savings can degrade the most, while datasets with much smaller savings can improve by several points. The central empirical fact is therefore not that VTC saves tokens. It clearly

[*]Equal contribution
[†]Corresponding authors

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Performance and token savings of the text path (LLM) versus the visual path (VLM) on 10 representative datasets using each task's native metric, evaluated with the Qwen3-4B model [40].

Dataset	Task type	Metric	Score		Token savings (%)
			LLM	VLM	
Visual path wins					
HotpotQA [49]	Multi-hop QA	F1	13.1	35.4	45.0
SciFact [43]	Fact Verification	F1	51.3	71.4	20.7
MuSiQue [41]	Multi-hop QA	F1	8.0	19.0	39.3
Yahoo Answers [51]	Topic Classification	Acc.	54.3	64.1	52.4
DBpedia [51]	Ontology Classification	Acc.	88.7	93.8	35.5
Text path wins					
ReCoRD [50]	Cloze-style QA	F1	51.3	26.5	12.7
DREAM [39]	Multi-choice RC	Acc.	88.0	71.1	12.8
IMDB [30]	Sentiment	Acc.	94.6	82.4	56.0
LogiQA [25]	Logical Reasoning	Acc.	50.2	38.9	12.8
CNN/DM [15]	Summarization	Rouge-1	34.0	24.1	68.6

does. The central problem is that the same rendering-and-encoding pipeline helps some tasks and hurts others, with no principled way to predict which case applies.

Existing reports suggest the same instability but stop short of explaining it. VTCBench [52] documents deficits in associative reasoning and long-term memory, while PixelWorld [29] reports pronounced drops on mathematics. Performance can collapse under semantic disruption [24], and simple mean pooling can rival vision-based compression on token-level reconstruction [22]. Together, these findings show that compression ratio is only an efficiency statistic: it tells us how many tokens are removed, but not what information is discarded, or when the visual path should be trusted.

Our starting point is that this failure of compression ratio is not accidental. Compression ratio measures savings, not task-relevant information loss. What is missing is a structured quantity that explains how the visual encoder changes the information available to the downstream task. We show that such a quantity arises naturally in the language of measure transport. After text is rendered, each textual unit obtains a spatial support through its glyph region in the image. Treating these spatially localized textual units and the resulting visual tokens as empirical probability measures, the ViT patch mechanism is not merely an opaque compressor. It can be modeled as a push-forward map that merges nearby textual units into coarser visual units and changes the geometry in which downstream reasoning is performed. This is the central analytical move of the paper. It turns VTC from an empirical compression trick into an object whose cost can be defined and estimated.

From this perspective, the transport cost induced by visual encoding admits a principled decomposition into two terms. The first is a precision cost induced by within-patch aggregation, which smooths fine-grained lexical distinctions. The second is a coverage cost induced by cross-patch fragmentation, which disperses evidence that a task may need to integrate across positions or pages. Both terms are estimable from downstream-label-free probes. Together, they provide the missing explanatory quantity behind the contradictions in Tab. 1. This transport-cost model does more than explain the phenomenon. Because the cost is label-free and input-computable, it yields operational tools directly. We derive a downstream-label-free decision criterion that selects whether the visual path should be used for a given input or task instance. We also derive a transport-informed foveation mechanism that identifies high-cost regions and selectively re-encodes them at higher resolution. The routing rule and the foveation module are therefore not separate heuristics. They are two consequences of the same transport-theoretic formulation. Our contributions are listed as:

- We show that VTC exhibits strong task-dependent instability across NLP benchmarks, and that compression ratio is an efficiency statistic rather than a predictor of downstream utility.

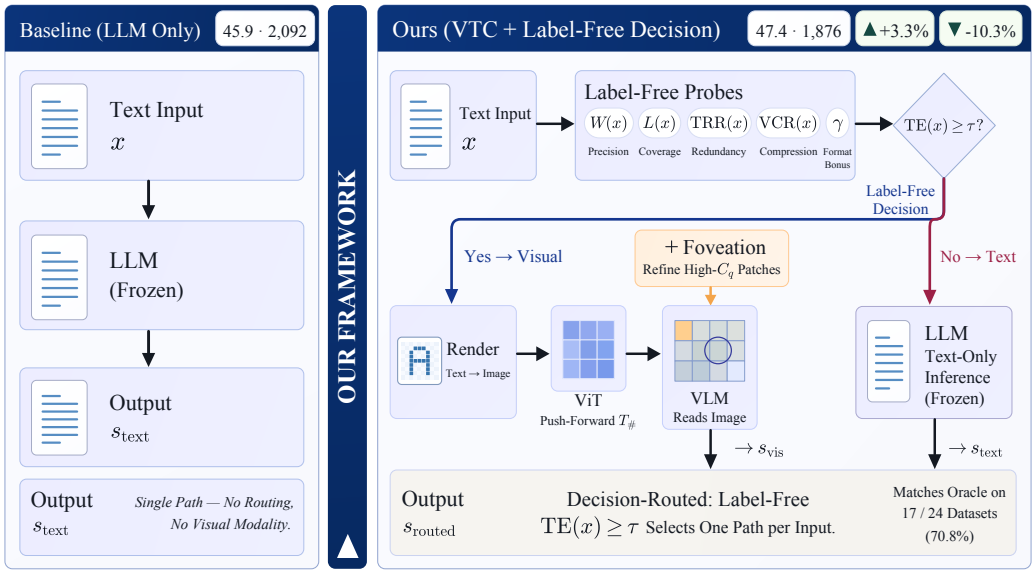


Figure 1: *Left*: the text-only baseline. *Right*: our VTC framework. Given x, we first compute label-free probes $(W, L, \text{TRR}, \text{VCR}, \gamma)$, combine them into the transport-efficiency score $\text{TE}(x)$, and route the instance under the rule $\text{TE}(x) \geq \tau$ (Sec. 3.2). If the visual path is selected, x is rendered to an image, encoded by the ViT push-forward map, and read by a VLM augmented with foveation that re-encodes high-C_q patches (Sec. 3.3). Across the 24-benchmark suite, the label-free rule matches the per-dataset oracle on $17/24$ datasets (70.8%), raises the macro-average score from 45.9 to 47.4 ($+1.5$ points, $+3.3\%$ relative), and uses 10.3% fewer average input tokens than pure-LLM.

- We provide a transport-theoretic account of this instability by recasting ViT-based visual encoding as a push-forward map and deriving a decomposed transport cost with precision and coverage terms, both estimable without downstream labels.
- We show that the cost model yields downstream-label-free routing and transport-informed foveation on 24 benchmarks, where routing improves the accuracy-token tradeoff over a pure-LLM baseline and foveation provides bounded local repair within the visual path.

2 Related Work

Recent works treat visual re-encoding as a practical interface for reducing textual sequence length. Frameworks like DeepSeek-OCR [9] and Glyph [46] instantiate this idea for long-context modeling, while later variants adapt it to dense retrieval, streaming, and mixed-modality inputs [18, 23, 47, 48]. The same pattern has also been repurposed beyond plain document ingestion, including long-context understanding [28], CoT compression [7, 45], and agent-memory compression [14]. A separate line of work probes the failure modes of this paradigm. VTCBench [52] surfaces deficits in associative reasoning and long-term memory, PixelWorld [29] reports weakness on mathematics, semantic perturbation studies expose brittleness beyond OCR fidelity [24], and reconstruction analyses show that naive pooling can rival vision-based compression on some token-level objectives [22]. What remains missing across these lines is not another application or stress test, but a structural account of the ViT-induced loss itself. Our method fills this gap by modeling visual encoding as a push-forward map and deriving a label-free transport cost that predicts when compression remains usable. Fig. 1 summarizes the resulting framework-level view of this formulation.

3 Method

We begin by formalizing VTC as a push-forward map induced by the rendering-and-encoding pipeline. For a text input x, the text path produces subword tokens $(x_1, \ldots, x_n)$. Rendering assigns each token

x_i a spatial glyph support in the image. The input therefore defines an empirical text measure

$$\mu_x \;=\; \frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}. \tag{1}$$

The visual path renders x and encodes the image with a ViT [12], producing visual tokens $(z_1, \ldots, z_m)$ with $m \ll n$. We view this pipeline as a many-to-one map T from localized textual units to visual-token supports. For any measurable set B in visual-token space, T induces the push-forward map

$$(T_{\#}\mu_x)(B) \;=\; \mu_x(T^{-1}(B)). \tag{2}$$

In the decoder-visible token view, suppressing preimage weights gives

$$\nu_x \;\approx\; \frac{1}{m}\sum_{j=1}^{m}\delta_{z_j}, \tag{3}$$

which keeps the visual-token support seen by the decoder. This rendering-induced push-forward reduces token count but changes the downstream information geometry. We therefore need a cost measuring task-relevant loss from replacing μ_x with ν_x.

3.1 Operationalized Transport Cost $C(x)$

A natural choice for quantifying the transport cost induced by $T_{\#}$ is the p-Wasserstein distance [35],

$$W_p(\mu_x, \nu_x) \;=\; \inf_{\gamma\in\Pi(\mu_x,\nu_x)}\left(\int \|u-v\|^p \, d\gamma(u,v)\right)^{1/p}, \tag{4}$$

which measures the minimum total cost of rearranging μ_x into ν_x under a chosen ground metric $\|u-v\|^p$, over all couplings $\gamma \in \Pi(\mu_x, \nu_x)$ with marginals μ_x and ν_x. This is the right mathematical object for transport, but it is not yet the right object for deployment. Three obstacles prevent us from adopting W_p directly. First, it suffers from ***ground-metric circularity***. A ground metric that reflects task-relevant loss must already know which lexical distinctions matter, such as "2023" versus "2024", but identifying exactly those distinctions is the purpose of $C(x)$. Second, it suffers from ***task-agnosticity***. The same W_p value may be harmless for multiple-choice reading comprehension and catastrophic for extractive span prediction. Third, it provides ***no downstream-label-free target***. Any usable threshold on W_p would have to be fit from labelled degradation curves, which violates our downstream-label-free requirement that no downstream label be consulted during inference.

Operational target. We use the score gap between the text path and the visual path as an operational measure of the loss introduced by the visual encoding pipeline. Let $s_{\text{text}}(x)$ and $s_{\text{vis}}(x)$ denote the task score of the text path and the visual path on the same input x. The quantity

$$\Delta(x) \;\triangleq\; s_{\text{text}}(x) - s_{\text{vis}}(x) \tag{5}$$

measures the performance loss associated with replacing the text-token interface by the rendered visual-token interface. $\Delta(x)$ resolves the first two obstacles. It requires no ground metric, and it inherits task dependence directly from the task scores themselves. It does not resolve the third obstacle (***no downstream-label-free target***), because computing $\Delta(x)$ still requires labels on each path's output. We therefore treat $\Delta(x)$ as a label-requiring ground-truth target and introduce a label-free proxy, a function $C(x)$ that is computable from the input alone and approximates $\Delta(x)$ without running either path. During validation, labels are used only to verify that $C(x)$ tracks $\Delta(x)$. During inference, only the proxy $C(x)$ is consulted. We emphasize that the resulting $C(x)$ is an engineering operationalization of the transport-loss target $\Delta(x)$, not a numerical solution to the optimal-transport problem in Eq. 4. Its functional form is motivated by the push-forward map structure derived next, but its scalars are estimated from label-free probes rather than from solving W_p, so all transport-cost statements in this paper refer to the proxy rather than to a computed Wasserstein distance.

Structure of Push-forward Map. To derive $C(x)$, we exploit the structure of encoder rather than fitting an unconstrained predictor. A ViT encoder does not apply $T_{\#}$ as a single map. It factorizes as

$$T_{\#} \;=\; T_{\text{inter}} \circ T_{\text{intra}}, \tag{6}$$

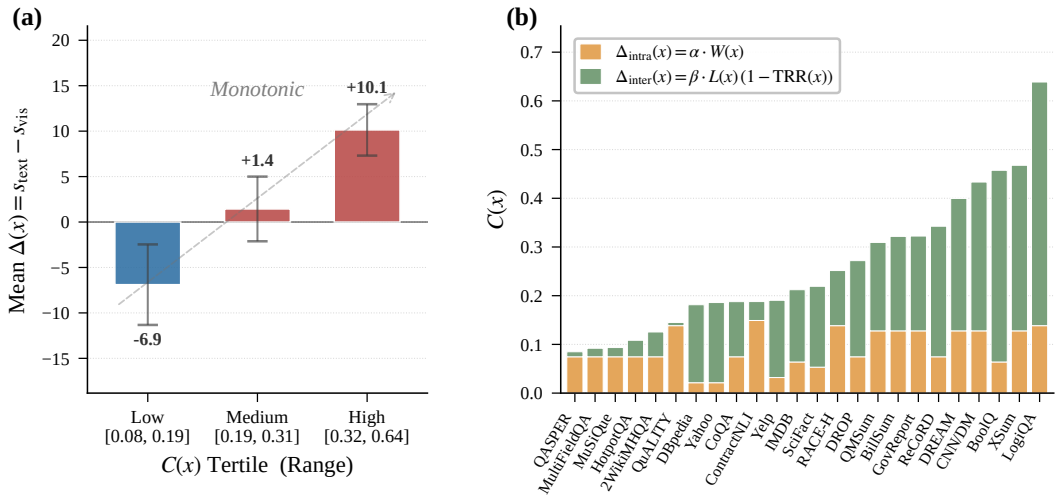


Figure 2: The label-free proxy $C(x)$ tracks the labelled operational gap $\Delta(x)$ on 24 NLP benchmarks with the Qwen3-4B backbone (LLM and VLM). **(a)** Mean $\Delta(x) = s_{\text{text}} - s_{\text{vis}}$ within each $C(x)$ tertile, with 8 datasets per bin and error bars of ± 1 standard error. Δ is reported in the native task metric, all on the same 0 to 100 scale. The mean Δ increases monotonically from -6.9 in the low-C tertile to $+10.1$ in the high-C tertile, showing that the label-free proxy orders the $C(x)$ tertiles consistently with the labelled operational gap. **(b)** Per-dataset decomposition of $C(x)$ into its intra-patch component $\Delta_{\text{intra}}(x) = \alpha \cdot W(x)$ (Orange) and inter-patch component $\Delta_{\text{inter}}(x) = \beta \cdot L(x)\,(1 - \text{TRR}(x))$ (Green), matching Eq. 8. The composition varies systematically with task structure. Δ_{inter} is largest on long-document summarization such as CNN/DM [15], XSum [32], GovReport [17], and BillSum [20], and on several high-coverage reasoning or classification inputs such as LogiQA [25] and BoolQ [10]. Δ_{intra} remains substantial on precision-sensitive tasks such as QuALITY [34], RACE-H [21], ContractNLI [19], and IMDB [30]. This empirical separation is consistent with the factorization $T_{\#} = T_{\text{inter}} \circ T_{\text{intra}}$ in Eq. 6.

where T_{intra} performs local aggregation by merging the spatially localized textual units inside each image patch into one patch embedding, and T_{inter} performs global arrangement by organizing the resulting patch embeddings under a 2D positional scheme. In this second stage, attention replaces the 1D linear order available to the text path. This factorization induces two distinct failure modes. The first is ***within-patch smoothing***. Fine-grained lexical distinctions inside the same patch, such as "2023" versus "2024", collapse into a shared averaged embedding. This harms tasks that require exact lexical fidelity. The second is ***cross-patch fragmentation***. Evidence distributed across patches, or across rendered pages for long inputs, must now be recomposed through 2D spatial attention rather than the native 1D positional structure of the text path. This weakens long-range evidence integration.

Because these two distortions act at different scales, we model the total transport loss additively:

$$\Delta(x) \;\approx\; \Delta_{\text{intra}}(x) + \Delta_{\text{inter}}(x), \tag{7}$$

with each term depending only on the corresponding stage of the push-forward map.

Label-free Proxy Function. Eq. 7 specifies the structure of the loss. We now instantiate it as a computable proxy. Each term is parameterized as the product of a label-free feature that measures sensitivity to the corresponding distortion and a model-specific scalar weight that measures how strongly that distortion hurts a given backbone. This yields

$$C(x) \;=\; \underbrace{\alpha \cdot W(x)}_{\Delta_{\text{intra}}(x)} \;+\; \underbrace{\beta \cdot L(x) \cdot \big(1 - \text{TRR}(x)\big)}_{\Delta_{\text{inter}}(x)}. \tag{8}$$

The precision feature $W(x) \in [0, 1]$ measures sensitivity to token-level distinctions in the required answer. It is high for extractive, generative, numerical, or option-level decisions where small lexical differences can change correctness, and low for coarse semantic classifications whose labels are robust to local lexical smoothing. $W(x)$ is determined from the task specification alone, without consulting any sample's ground-truth label.

The coverage feature $L(x) \in [0, 1]$ measures how widely task-relevant evidence is distributed across the rendered input. To estimate it, we compute BM25 [37] relevance scores between the question and text segments, normalize these scores into a relevance distribution over segments, and summarize the dispersion of this relevance mass. Higher $L(x)$ indicates that task-relevant evidence is spread across more segments, making it more vulnerable to the 2D rearrangement induced by T_{inter}. The text redundancy rate $\mathrm{TRR}(x) \in [0, 1]$, estimated by the gzip compression ratio [42] of x, discounts the coverage term. When the text is highly redundant, duplicate carriers of the same information buffer fragmentation loss, so only the irreducible fraction $1 - \mathrm{TRR}(x)$ contributes to the cost.

All three features are available before any downstream label is observed. Scalar weights α and β are model-specific because the severity of each distortion depends on the ViT architecture and its coupling to the decoder. They are estimated from independent synthetic probes and never see downstream-task labels. $C(x)$ is therefore fully label-free at deployment. $\Delta(x)$ enters only as the validation target against which $C(x)$ is checked, never as an input to inference.

Empirical Check. Fig. 2 verifies that the derived proxy tracks the labelled operational gap on the benchmark suite. Panel **(a)** groups 24 benchmarks into $C(x)$ tertiles and shows that the mean gap $\Delta(x)$ increases monotonically from the low-C bin to the high-C bin. This confirms that $C(x)$ orders the tertiles consistently with $\Delta(x)$ without consulting downstream labels. Panel **(b)** decomposes $C(x)$ for every dataset and shows that the inter-patch term is largest on long-document and high-coverage inputs, while precision-sensitive tasks retain substantial intra-patch cost. The empirical split is consistent with the theoretical factorization in Eq. 6, indicating that the two terms in $C(x)$ correspond to distinct failure modes of the ViT-induced push-forward map rather than to compression ratio alone.

3.2 Path Selection via Transport Efficiency (TE)

Path selection must balance the transport loss measured by $C(x)$ against token savings from visual compression. We define a TE score that combines surviving information with compression benefit.

Compression Benefit and Structure Bonus. The benefit term is the Visual Compression Ratio

$$\mathrm{VCR}(x) \triangleq \frac{n(x)}{m(x)}, \tag{9}$$

where $n(x)$ is the number of text tokens produced by subword tokenization and $m(x)$ is the number of visual tokens produced by the ViT. $\mathrm{VCR}(x) > 1$ means rendering shortens the sequence the decoder must process. A second effect is specific to structured inputs. When x contains tabular or list layout, rendering preserves 2D spatial relations that a flat token sequence cannot express, and the visual path can exploit these relations at decode time. We summarize this effect by a nonnegative structured-format bonus γ, estimated from an independent probe.

TE. We normalize the text path to a baseline information value of 1. Relative to that baseline, the visual path loses information through cost $C(x)$ and may gain information through the structured-format bonus γ. We call the resulting quantity the Information Survival Rate,

$$\mathrm{ISR}(x) \triangleq 1 + \gamma - C(x), \tag{10}$$

which is below 1 when transport loss dominates and above 1 when the layout bonus dominates. Multiplying surviving information by compression benefit gives the TE,

$$\begin{aligned} \mathrm{TE}(x) &\triangleq \mathrm{ISR}(x) \cdot \mathrm{VCR}(x) \\ &= \big(1 + \gamma - C(x)\big) \cdot \mathrm{VCR}(x), \end{aligned} \tag{11}$$

which measures the net value of selecting the visual path for x. High $\mathrm{TE}(x)$ means that the visual path preserves enough information to justify its token savings. Low $\mathrm{TE}(x)$ means that the push-forward map destroys more information than the compression benefit can repay.

Routing then reduces to a single threshold:

$$\text{select the visual path for } x \iff \mathrm{TE}(x) \geq \tau, \tag{12}$$

where τ is a fixed decision margin that absorbs residual uncertainty in the probe-based parameter estimates. The parameters (α, β, γ) are model-specific and estimated by the probes.

Decision geometry. According to Eq. 11, the threshold condition $\mathrm{TE}(x) \geq \tau$ traces a hyperbola in the $(\mathrm{VCR}, \mathrm{ISR})$ plane:

$$\mathrm{ISR}(x) = \frac{\tau}{\mathrm{VCR}(x)}.$$

This hyperbola partitions the plane into a select-visual region above the curve and a select-text region below it. Fig. 3 visualizes this partition on the 24-benchmark suite at Qwen3-4B scale. Each dataset's position relative to the curve encodes the trade-off between transport loss and compression benefit. The visual separation allows a single scalar τ, shared across all 24 benchmarks at this scale, to carve out the two regimes without per-benchmark tuning. The 4B operating point $\tau = 1.28$ is fixed before downstream evaluation. App. F reports only post-hoc sensitivity diagnostics and cross-scale behavior.

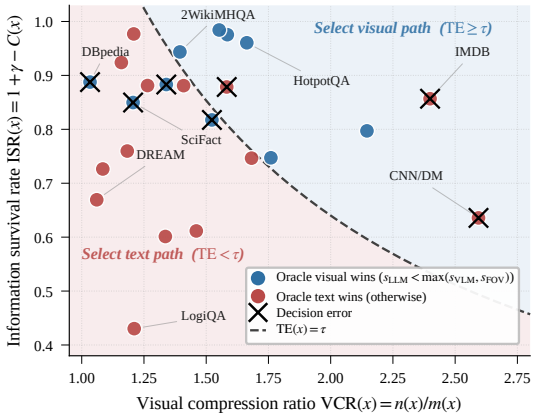


Figure 3: TE decision plane at 4B scale. Each of 24 benchmarks is placed at its $(\mathrm{VCR}(x), \mathrm{ISR}(x))$ coordinates and coloured by the oracle preference. The dashed curve is the $\mathrm{TE}(x) = \tau$ contour, a hyperbola because $\mathrm{TE}(x) = \mathrm{ISR}(x) \cdot \mathrm{VCR}(x)$. The two shaded regions are the select-visual and select-text assignments of the rule $\mathrm{TE}(x) \geq \tau$, and decision errors are marked with an $\times$. For oracle labeling, the visual arm is the better of VLM and VLM+Fov, matching the upper-bound definition in Tab. 2. The single scalar τ is fixed across all benchmarks. The rule is correct on $17/24$ datasets (70.8%) at Qwen3-4B, raises the macro-average score from 45.9 to 47.4 $(+1.5$ points, $+3.3\%$ relative), and uses 10.3% fewer average input tokens than the LLM-only baseline model.

3.3 Localized Inverse Transport

The scalar $C(x)$ decides whether the visual path is viable. It does not assume that transport loss is spatially uniform. In practice, some patches survive aggregation gracefully, while others concentrate exactly the distinctions that T_{intra} washes out. For inputs already routed to the visual path, we partially recover this local loss by re-encoding selected regions at finer resolution and appending the additional tokens to the base visual measure. We call this foveation. In the transport view, foveation is a localized inverse of the push-forward map:

$$\nu_{\text{fov}} = \nu_{\text{base}} + \sum_k \Delta\nu_k. \tag{13}$$

The remaining questions are where to refine, when refinement is worthwhile, and how the refinement mechanism relates to the same cost model used for routing.

Transport cost C_q. We spatialize the global proxy by defining, for each patch q in rendered image,

$$C_q = \alpha \cdot W_q + \beta \cdot L_q \cdot (1 - \mathrm{TRR}_q), \tag{14}$$

using same weights α and β as in Eq. 8, together with patch-level features that mirror their input-level counterparts. W_q is the normalized variance of ViT features inside patch q. Higher variance indicates more local detail that T_{intra} is likely to smooth out. L_q is the local contribution of patch q to the segment-level BM25 relevance distribution used for $L(x)$. Higher L_q marks patches that are more likely to contain task-relevant evidence. TRR_q is the gzip compression ratio of the text rendered inside patch q. It discounts patches whose content is redundantly available elsewhere in the document.

The text-to-patch alignment is deterministic. Rendering is a fixed function of x with known fonts, page width, and line spacing, so each line of text occupies a known pixel region and maps into a known set of ViT patches. No learned alignment is required. The result is a cost map $\{C_q\}$ in which regions concentrating task-relevant lexical detail or dense evidence appear as hot spots.

Foveation trigger. Foveation is useful only if the information it recovers exceeds the extra token cost it introduces. Suppose we select a set $\{k\}$ of regions, yielding a total cost recovery $\sum_k \Delta C_k$ at the price of n_c additional visual tokens. The post-foveation transport efficiency is

$$\mathrm{TE}_{\text{fov}}(x) = \left(\mathrm{ISR}(x) + \sum_k \Delta C_k\right) \cdot \frac{n_t}{n_v + n_c}, \tag{15}$$

Table 2: Main results at the Qwen3-4B scale. Scores are macro-averages of per-dataset metrics (Tab. 5 in App. C). Tokens are the mean input-token count per sample. ***Oracle-routed*** picks the better of LLM and the best visual arm post hoc and gives an upper bound (UB).

	Score				
Method	S2	S3	All	Tokens	Δ vs LLM
LLM	34.7	54.0	45.9	2,092	—
VLM	44.0	44.6	44.4	1,486	-1.5 / -29.0%
VLM + Foveation	44.7	44.8	44.8	1,521	-1.1 / -27.3%
Decision-routed (Ours)	41.1	51.9	**47.4**	**1,876**	$\mathbf{+1.5}$ / $\mathbf{-10.3\%}$
Oracle-routed (UB)	44.7	54.0	*50.2*	*1,894*	$+4.3$ / -9.5%

where n_t and n_v are the text-path and base visual-path token counts. Requiring $\mathrm{TE}_{\mathrm{fov}}(x) > \mathrm{TE}(x)$ and simplifying gives

$$\frac{\sum_k \Delta C_k}{\mathrm{ISR}(x)} > \frac{n_c}{n_v}. \tag{16}$$

The left-hand side is the fraction of surviving information recovered by foveation. The right-hand side is the token overhead relative to the base visual encoding. This inequality is the sole trigger for refinement. It inherits all of its parameters from Sec. 3.1 and Sec. 3.2.

Region selection. Given the trigger, we still need to choose which patches to refine. We sort patches by C_q in descending order, apply non-maximum suppression at a fixed spatial radius to avoid overlapping crops, and add regions greedily until either Eq. 16 is satisfied or the crop budget is exhausted. The crop budget caps extra tokens at $n_c \leq 0.25 \cdot n_v$, a fixed engineering constant used in all experiments. If the greedy procedure terminates before the trigger is met, foveation is skipped and the base visual encoding is used unchanged.

4 Experiments

4.1 Experimental Setup

We evaluate four inference strategies on the 24-benchmark suite of Sec. C: LLM (text-only), VLM (text rendered to an image), VLM+Fov (visual path with localized inverse transport, Sec. 3.3), and Decision-routed, which selects a path using $\mathrm{TE}(x) \geq \tau$ with $\tau = 1.28$ fixed before downstream evaluation (Sec. 3.2). Downstream outcomes are used only for the post-hoc sensitivity analysis in App. F, not to choose the reported operating point. We also report an Oracle-routed upper bound (UB) that picks the better of LLM and the best visual arm post hoc. We use the Qwen3 Instruct model series as the shared backbone because it provides LLMs and VLMs at matched parameter counts, making two paths directly comparable. We run experiments at 4B, 8B, and 32B. ***Due to space constraints, the main text reports the 4B results, while the 8B and 32B results are deferred to the App. E.*** All experiments are run on NVIDIA L40S GPUs. Scores are per-dataset means of the metric listed in Tab. 5 (App. C). Tokens are mean input tokens per sample. We report macro-averages within the two scenarios S2 (visual-friendly, $n=10$) and S3 (text-friendly, $n=14$).

4.2 Main Results

Tab. 2 reports the 4B results, aggregated over S2, S3, and all 24 benchmarks. ***The per-dataset breakdown and finer-grained analysis stratified by scenario sub-types are deferred to the appendix.*** The discussion below is organized around four claims that match the theoretical structure of Sec. 3.

Claim 1: Naive VTC has task-dependent transport cost. Across all 24 benchmarks, VLM scores 1.5 points below LLM on average (44.4 vs. 45.9) while consuming 29.0% fewer tokens. The gap is highly non-uniform. On S2, the visual path outperforms LLM by $+9.3$ points, showing that rendering can preserve useful structure when relevant evidence remains visually recoverable. On S3, the visual path falls 9.4 points below LLM, consistent with the loss of fine-grained lexical and logical cues

under within-patch aggregation. This confirms the central prediction of Sec. 3: the ViT push-forward map is not uniformly beneficial or harmful, but induces a task-dependent transport cost.

Claim 2: Foveation recovers part of the transport cost but is insufficient alone. Adding localized inverse transport (Sec. 3.3) lifts the visual path from 44.4 to 44.8, reducing the average deficit to LLM by 0.4 points at $<3\%$ extra token cost ($1{,}521$ vs. $1{,}486$). The gains concentrate where a few high-C_q patches carry most of the useful evidence ($+0.7$ on S2) and fade where the cost is more diffuse ($+0.2$ on S3). Foveation therefore complements the visual path by locally repairing high-cost regions, but it cannot replace a global path-selection decision.

Claim 3: Label-free routing converts the two paths into a net win. Our Decision-routed strategy, selecting the visual arm iff $\mathrm{TE}(x) \geq \tau$ with a single $\tau = 1.28$ fixed a priori, matches the oracle choice on $17/24$ datasets (70.8%; Fig. 3) and achieves $+1.5$ average score over LLM-only at -10.3% tokens, without consulting any training labels at inference. The gain splits as expected: on S3, the rule preserves 96% of LLM's score while still reducing tokens by about 2%; on S2, it captures 64% of the visual-path upside relative to the oracle.

Claim 4: A 2.8-point headroom remains to the oracle. The Oracle-routed row shows the label-aware upper bound (UB) at $+4.3$ points over LLM, while our label-free rule reaches $+1.5$ points and captures about 35% of this gain. The remaining 2.8-point gap is concentrated in borderline cases where the current proxy under-estimates structural loss or benefit. This suggests that the cost model captures the dominant routing signal, but finer sample-level structure could further close the gap.

4.3 Ablations and Sensitivity

Our proposed label-free rule can match the supervised predictor. Tab. 3 compares our downstream-label-free OT rule against supervised baselines trained leave-one-dataset-out (LOOCV) across 24 benchmarks, with the per-dataset oracle ("visual arm beats LLM") as the binary label. Single-feature rows pick the best label-fitting threshold on the listed feature. The 4-feature classifiers take $(W, L, \mathrm{TRR}, \mathrm{VCR})$. The 5-feature row adds LLM input length. Our rule reaches 70.8% ***without labels***, matching the best supervised method (5-feature LogReg) and showing that the transport-cost structure pre-encodes most of the label-selection signal. The 4-feature LogReg ties single-feature thresholds at 66.7% because the four probes carry the same per-dataset rank as VCR alone. The fifth feature lifts accuracy by distinguishing long-context inputs whose transport benefit exceeds compression-ratio alone. Decision tree (depth 3) and MLP underperform because 24-sample LOOCV gives too little supervision for axis-aligned splits or non-linear models to beat a linear discriminant.

Table 3: ***Path-selection predictors at 4B***. Accuracy vs. the per-dataset oracle on 24 benchmarks. ***Labels*** marks whether the predictor sees task labels. Supervised rows use LOOCV.

Predictor	Labels	Acc.
VCR-only (best θ)	✓	66.7%
W-only (best θ)	✓	66.7%
LogReg ($W, L, \mathrm{TRR}, \mathrm{VCR}$)	✓	66.7%
LogReg (5 features)	✓	$\mathbf{70.8\%}$
DecisionTree (depth 3)	✓	37.5%
MLP (hidden 16)	✓	50.0%
SVM-RBF	✓	58.3%
OT rule (ours, label-free)	✗	$\mathbf{70.8\%}$

Table 4: Rendering robustness at the Qwen3-4B scale for three font sizes.

Font size	Scores	Tokens
font10	42.4	1,185
font12 (Default)	44.4	1,486
font14	45.3	1,716

The framework shows predictable rendering sensitivity. We sweep font size in $\{10, 12, 14\}$ at the Qwen3-4B scale while holding other rendering parameters fixed (Tab. 4). The macro score changes monotonically with font size: shrinking from 12 pt to 10 pt drops the score by 2.0 points and reduces mean input tokens by 20%, while enlarging to 14 pt raises the score by 0.9 points at a roughly 16% token cost. The total variation is about 3 points across the sweep and shows no direction reversal, indicating that the visual path follows a consistent score-token tradeoff rather than relying on a tuned rendering setting. This direction is consistent with the push-forward map: smaller fonts pack more characters into each ViT patch and therefore amplify lossy within-patch aggregation. Full details of the rendering pipeline are provided in App. H.

5 Conclusion

We cast visual text compression as a measure-transport problem and showed that the ViT push-forward map decomposes into precision and coverage costs. These costs are estimated from downstream-label-free probes, yielding a scalar transport-efficiency score $\mathrm{TE}(x)$ whose threshold defines a label-free path-selection rule. The same cost model extends to the patch level through C_q, motivating bounded foveation that re-encodes only costly regions. Because global routing and local refinement share the same probe-derived cost parameters (α, β, γ), while routing alone introduces the decision threshold τ, the downstream-label-free design is preserved across both stages. On 24 NLP benchmarks at the 4B scale, the rule matches the per-dataset oracle on $17/24$ datasets (70.8%), improves the macro-average score by $+1.5$ points ($+3.3\%$ relative), uses 10.3% fewer average input tokens than a pure-LLM baseline, and matches the best supervised predictor without downstream labels. These results suggest that VTC's task-dependent contradictions arise from non-uniform transport cost rather than compression ratio alone. We hope that the modeling perspective and experimental design developed here can provide useful starting points for future research on visual text compression.

6 Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. T2588101, 52572334) and Xishan-Tsinghua Special Funding for the Deep Integration of Industry, Academia and Research. This work was also supported in part by the Canada CIFAR AI Chairs Program, the Alberta Machine Intelligence Institute (Amii), and the Natural Sciences and Engineering Research Council of Canada (NSERC)

References

[1] Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9392–9401, 2025.

[2] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding. In *ACL (1)*, pages 3119–3137. Association for Computational Linguistics, 2024.

[3] Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. Lessons from the trenches on reproducible evaluation of language models. *CoRR*, abs/2405.14782, 2024.

[4] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. PLOT: prompt learning with optimal transport for vision-language models. In *ICLR*. OpenReview.net, 2023.

[5] Lingyi Chen, Shitong Wu, Wenhao Ye, Huihui Wu, Hao Wu, Wenyi Zhang, Bo Bai, and Yining Sun. Information bottleneck revisited: Posterior probability perspective with optimal transport. In *2023 IEEE International Symposium on Information Theory (ISIT)*, pages 1490–1495. IEEE, 2023.

[6] Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. Graph optimal transport for cross-domain alignment. In *ICML*, Proceedings of Machine Learning Research, pages 1542–1553. PMLR, 2020.

[7] Xiaoshu Chen, Sihang Zhou, Ke Liang, Taichun Zhou, and Xinwang Liu. Imgcot: Compressing long chain of thought into compact visual tokens for efficient reasoning of large language model. *CoRR*, abs/2601.22730, 2026.

[8] Xiwen Chen, Wenhui Zhu, Gen Li, Xuanzhao Dong, Yujian Xiong, Hao Wang, Peijie Qiu, Qingquan Song, Zhipeng Wang, Shao Tang, et al. Otprune: Distribution-aligned visual token pruning via optimal transport. *arXiv preprint arXiv:2602.20205*, 2026.

[9] Jiale Cheng, Yusen Liu, Xinyu Zhang, Yulin Fei, Wenyi Hong, Ruiliang Lyu, Weihan Wang, Zhe Su, Xiaotao Gu, Xiao Liu, Yushi Bai, Jie Tang, Hongning Wang, and Minlie Huang. Glyph: Scaling context windows via visual-text compression. *CoRR*, abs/2510.17800, 2025.

[10] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL-HLT (1)*, pages 2924–2936. Association for Computational Linguistics, 2019.

[11] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *NAACL-HLT*, pages 4599–4610. Association for Computational Linguistics, 2021.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021.

[13] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL-HLT (1)*, pages 2368–2378. Association for Computational Linguistics, 2019.

[14] Lang Feng, Fuchao Yang, Feng Chen, Xin Cheng, Haiyang Xu, Zhenglin Wan, Ming Yan, and Bo An. Agentocr: Reimagining agent history via optical self-compression. *CoRR*, abs/2601.04786, 2026.

[15] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *NeurIPS*, pages 1693–1701, 2015.

[16] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *COLING*, pages 6609–6625. International Committee on Computational Linguistics, 2020.

[17] Luyang Huang, Shuyang Cao, Nikolaus Nova Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization. In *NAACL-HLT*, pages 1419–1436. Association for Computational Linguistics, 2021.

[18] Dian Jiao, Jiaxin Duan, Shuai Zhao, Jiabing Leng, Yiran Zhang, and Feng Huang. Global context compression with interleaved vision-text transformation. *CoRR*, abs/2601.10378, 2026.

[19] Yuta Koreeda and Christopher D. Manning. Contractnli: A dataset for document-level natural language inference for contracts. In *EMNLP (Findings)*, Findings of ACL, pages 1907–1919. Association for Computational Linguistics, 2021.

[20] Anastassia Kornilova and Vlad Eidelman. Billsum: A corpus for automatic summarization of US legislation. *CoRR*, abs/1910.00523, 2019.

[21] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. RACE: large-scale reading comprehension dataset from examinations. In *EMNLP*, pages 785–794. Association for Computational Linguistics, 2017.

[22] Ivan Yee Lee, Cheng Yang, and Taylor Berg-Kirkpatrick. Optical context compression is just (bad) autoencoding. *CoRR*, abs/2512.03643, 2025.

[23] Yanhong Li, Zixuan Lan, and Jiawei Zhou. Text or pixels? it takes half: On the token efficiency of visual text inputs in multimodal llms. *CoRR*, abs/2510.18279, 2025.

[24] Yunhao Liang, Ruixuan Ying, Bo Li, Hong Li, Kai Yan, Qingwen Li, Min Yang, Okamoto Satoshi, Zhe Cui, and Shiwen Ni. Visual merit or linguistic crutch? A close look at deepseek-ocr. *CoRR*, abs/2601.03714, 2026.

[25] Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *IJCAI*, pages 3622–3628. ijcai.org, 2020.

[26] Tianlin Liu, Mathieu Blondel, Carlos Riquelme, and Joan Puigcerver. Routers in vision mixture of experts: An empirical study. *arXiv preprint arXiv:2401.15969*, 2024.

[27] Xugang Lu, Peng Shen, Yu Tsao, and Hisashi Kawai. Cross-modal alignment with optimal transport for ctc-based ASR. In *ASRU*, pages 1–7. IEEE, 2023.

[28] Yujie Lu, Xiujun Li, Tsu-Jui Fu, Miguel P. Eckstein, and William Yang Wang. From text to pixel: Advancing long-context understanding in mllms. *CoRR*, abs/2405.14213, 2024.

[29] Zhiheng Lyu, Xueguang Ma, and Wenhu Chen. Pixelworld: How far are we from perceiving everything as pixels? *CoRR*, abs/2501.19339, 2025.

[30] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *ACL*, pages 142–150. The Association for Computer Linguistics, 2011.

[31] Shweta Mahajan, Teresa Botschen, Iryna Gurevych, and Stefan Roth. Joint wasserstein autoencoders for aligning multimodal embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[32] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *EMNLP*, pages 1797–1807. Association for Computational Linguistics, 2018.

[33] Duc Anh Nguyen, Huu Binh Ta, Nhuan Le Duc, Tan M Nguyen, and Toan Tran. Selective sinkhorn routing for improved sparse mixture of experts. *arXiv preprint arXiv:2511.08972*, 2025.

[34] Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel R. Bowman. Quality: Question answering with long input texts, yes! In *NAACL-HLT*, pages 5336–5358. Association for Computational Linguistics, 2022.

[35] Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Found. Trends Mach. Learn.*, 11(5-6): 355–607, 2019.

[36] Siva Reddy, Danqi Chen, and Christopher D. Manning. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7:249–266, 2019.

[37] Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009.

[38] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020.

[39] Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. Dream: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231, 2019.

[40] Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025.

[41] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554, 2022.

[42] Paul M. B. Vitányi, Frank J. Balbach, Rudi Cilibrasi, and Ming Li. Normalized information distance. *CoRR*, abs/0809.2553, 2008.

[43] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *EMNLP (1)*, pages 7534–7550. Association for Computational Linguistics, 2020.

[44] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, JingJing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *CoRR*, abs/2508.18265, 2025.

[45] Yifan Wang, Shiyu Li, Peiming Li, Xiaochen Yang, Yang Tang, and Zheng Wei. Render-of-thought: Rendering textual chain-of-thought as images for visual latent reasoning. *CoRR*, abs/2601.14750, 2026.

[46] Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr: Contexts optical compression. *CoRR*, abs/2510.18234, 2025.

[47] Ling Xing, Alex Jinpeng Wang, Rui Yan, Hongyu Qu, Zechao Li, and Jinhui Tang. See the text: From tokenization to visual reading. *CoRR*, abs/2510.18840, 2025.

[48] Ling Xing, Alex Jinpeng Wang, Rui Yan, and Jinhui Tang. Vision-centric token compression in large language model. *CoRR*, abs/2502.00791, 2025.

[49] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, pages 2369–2380. Association for Computational Linguistics, 2018.

[50] Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. Record: Bridging the gap between human and machine commonsense reading comprehension. *CoRR*, abs/1810.12885, 2018.

[51] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NeurIPS*, pages 649–657, 2015.

[52] Hongbo Zhao, Meng Wang, Fei Zhu, Wenzhuo Liu, Bolin Ni, Fanhu Zeng, Gaofeng Meng, and Zhaoxiang Zhang. Vtcbench: Can vision-language models understand long context with vision-text compression? *CoRR*, abs/2512.15649, 2025.

[53] Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir R. Radev. Qmsum: A new benchmark for query-based multi-domain meeting summarization. In *NAACL-HLT*, pages 5905–5921. Association for Computational Linguistics, 2021.

[54] Xingyu Zhu, Beier Zhu, Shuo Wang, Kesen Zhao, and Hanwang Zhang. Enhancing clip robustness via cross-modality alignment. *arXiv preprint arXiv:2510.24038*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's scope as a measure-transport account of visual text compression, with downstream-label-free routing and transport-informed foveation. The corresponding evidence is reported in Secs. 3–4, Tab. 2, Fig. 1, and the diagnostic appendices.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: App. A explicitly summarizes scope and limitations, including the operational nature of $C(x)$, non-universal thresholds, rendering choices, foveation limits, and cross-architecture scope. Additional boundary cases are discussed in App. F, App. G, App. H, App. I, and App. J.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not state formal theorems or lemmas requiring proofs. It provides a modeling formulation and operational derivations in Sec. 3, and explicitly states in App. A that $C(x)$ is an engineering operationalization rather than a numerical solution to the optimal-transport problem.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sec. 4 and App. C specify the benchmark suite, sample counts, metrics, answer formats, and matched LLM/VLM prompts. App. D, App. E, App. H, App. I, and App. J document calibration, per-dataset results, rendering, hardware/runtime measurements, and cross-architecture settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The current manuscript documents the experimental protocol and uses public benchmark datasets and released model families, but it does not provide an anonymized public code/data URL or exact reproduction commands. The appendix states that full instruction text and output parsers will be released with the code.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The work does not train new models; it evaluates fixed Qwen3/Qwen3-VL and InternVL3.5 checkpoints. Sec. 4 and the appendices specify model scales, routing threshold, synthetic probe calibration, prompt templates, rendering parameters, dataset sample counts, metrics, and evaluation hardware.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Fig. 2 reports ± 1 standard-error bars for the diagnostic $C(x)$ tertile analysis, with each tertile containing 8 benchmarks. For the main benchmark tables, we report fixed-suite macro averages over 24 established datasets rather than repeated stochastic training runs. The evaluation uses fixed checkpoints, fixed prompts, and deterministic scoring protocols, so the primary source of variation is across benchmark datasets rather than random initialization or training variance.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Sec. 4 states that all experiments are run on NVIDIA L40S GPUs. App. I further reports representative wall-clock latency and peak GPU memory for the 4B LLM, VLM, VLM+Fov, and routed runs. The experiments use released checkpoints and offline inference rather than model training, so the dominant compute requirement is evaluation-time GPU inference. The appendix also documents the benchmark suite, rendering protocol, routing diagnostics, foveation settings, and cross-architecture setup needed to reproduce the reported runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: To the best of our knowledge, the work conforms to the NeurIPS Code of Ethics. It evaluates released language and vision-language model checkpoints on established benchmarks, does not collect new personal data, and does not involve human-subject studies or deployment decisions.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The manuscript focuses on methodological and empirical analysis and does not include a dedicated broader-impact discussion. The work may improve efficient long-context processing, but it also inherits the general risks of LLM/VLM deployment if used in downstream systems.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a new pre-trained model, image generator, scraped dataset, or other high-risk asset. It evaluates a routing/foveation method using existing public benchmarks and released model checkpoints.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper cites the datasets, model families, and related benchmarks used in the evaluation, including the 24 benchmark suite and Qwen3/Qwen3-VL/InternVL3.5. However, the current manuscript does not explicitly list each asset's license or terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce a new dataset, benchmark, or model asset as its main contribution. The synthetic probes are procedural evaluation/calibration tasks and are documented in App. D.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, new annotation work, user studies, or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: No IRB approval is required for this work because it does not involve human-subject experiments, crowdsourcing, or new data collection from people.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs and VLMs are central experimental components of the research, and the paper describes the text-only LLM path, the visual VLM path, Qwen3/Qwen3-VL model scales, InternVL3.5 cross-architecture validation, prompts, and evaluation protocols. This declaration concerns model usage in the research method.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

A Overview of Supplementary Experiments and Analyses

The purpose of this section: This section provides a roadmap for the appendix. The main paper presents the transport-cost framework, the label-free path-selection rule, and the headline 4B routing and foveation results. The appendix opens with a positioning of the framework against prior optimal-transport work in deep learning, then extends the empirical evidence by documenting the evaluation protocol, probe calibration, cross-scale behavior, foveation limitations, rendering sensitivity, runtime measurements, and cross-architecture validation. Together, these analyses are intended to support the ***reproducibility***, ***downstream-label-free design***, and ***transport-cost interpretation*** of the main results.

What is documented in the appendix. Beyond this overview, the supplementary material is organized into the following technical sections, each addressing a distinct concern that a careful reader may have after the main text.

App. B situates the measure-transport formulation within the broader use of optimal transport in deep learning, organizes prior work into four families (cross-modal alignment, visual-token selection, MoE routing, and information-bottleneck/compression), identifies the structural difference between each family and our framework, addresses two recent VTC critiques on the boundary of our claim, and summarizes what is and is not new in our contribution. The goal is to make explicit that our framework treats OT as something to diagnose, not optimize, so that a reader familiar with any one of these families does not read our work as a variant.

App. C documents the 24-benchmark suite, including dataset selection, native metrics, answer formats, and the unified prompt templates used for the LLM and visual paths. The goal is to make the headline numbers reproducible at the data layer and to rule out prompt design as a source of the observed LLM/VLM gap.

App. D reports the construction and fits of the three downstream-label-free synthetic probes used to estimate (α, β, γ). It also provides a sample-paired validation showing that the fixed cost ordering $C(x)$ predicts VLM-vs-LLM behavior on real evaluation samples. The goal is to show that the parameters underlying the rule are not fit on the 24 downstream benchmarks, and that the resulting cost is aligned with observed visual advantage after the cost has already been fixed.

App. E reports per-family theory-data validation and the full per-dataset grid at 4B, 8B, and 32B for LLM, VLM, and VLM with foveation. The goal is to verify that the qualitative regimes implied by the cost decomposition, including intra-dominated, inter-dominated, low-cost, and mixed regimes, appear at the family and dataset levels rather than only in macro-averages.

App. F reports threshold sensitivity, cross-scale routing diagnostics, and an external path-identification diagnostic on VTCBench. The goal is to show that the 4B operating point is not a fragile threshold choice, to diagnose how the same transport-efficiency framework behaves under scale-dependent probe regimes at 8B and 32B, and to test subtask-level path identification on an external benchmark not used in calibration.

App. G reports per-scale foveation effects, foveation trigger statistics, and qualitative case studies. The goal is to characterize when foveation helps, when it is limited, and how its scale dependence follows the same parameter regimes identified by the probes.

App. H documents the deterministic rendering protocol and reports a font-size sensitivity sweep at all three scales. The goal is to show that the rendering setup is fully specified, applied uniformly to every benchmark, and that the visual path responds to font size in the direction predicted by within-patch aggregation rather than depending on dataset-specific rendering choices.

App. I reports wall-clock latency and peak GPU memory for the four inference strategies on six representative datasets at the 4B scale. The goal is to verify that the token-count savings translate into measurable runtime savings, and that the decision-routed strategy does not introduce a runtime cost that erases its accuracy gains.

App. J reports a sample-paired cross-architecture replication of the 4B, 8B and 32B baseline experiments using InternVL3.5, which pairs an InternViT visual encoder with the same Qwen3 LLM backbones used in the headline runs. The goal is to test whether the framework's task-dependent visual-versus-text pattern is specific to the Qwen3-VL family or transfers to a different visual encoder, while keeping the LLM backbone matched and using the result as a cross-architecture diagnostic of the visual encoding stack.

What this paper does and does not claim. We do not claim a state-of-the-art result on any individual benchmark in this suite. Several included benchmarks have stronger task-specific systems in the literature, and our objective is not to compete with them at the leaderboard level. The contribution of this paper is upstream of any specific benchmark number. We recast visual text compression as a measure-transport problem, derive a structured transport cost with two separately estimable components, and use it to obtain a label-free path-selection rule and a transport-informed foveation mechanism that share their parameters. The experiments in the main text and appendix are designed to test whether this framework explains the task-dependent behavior of VTC, not to set a leaderboard. We hope that the modeling perspective and the supplementary evidence collected here provide a useful starting point for future work on visual text compression.

Scope and limitations. We summarize the deliberate scope choices that shape this paper and the resulting limitations a reader should keep in mind when interpreting the results.

First, the transport cost $C(x)$ is an engineering operationalization of the labelled gap $\Delta(x)$, not a numerical solution to the optimal-transport problem of Eq. 4. Its functional form is motivated by the push-forward map structure of Sec. 3, but its parameters are estimated from label-free probes rather than from a Wasserstein computation.

Second, the operating point $\tau = 1.28$ is selected at the Qwen3-4B scale and is not claimed as a universal cross-scale constant. The 8B and 32B columns of App. F are diagnostic, and report scale-specific thresholds where applicable.

Third, the decomposition $C(x) = \alpha W + \beta L(1 - \mathrm{TRR})$ is the simplest two-term proxy that the push-forward factorization motivates. We do not claim it is the only or optimal feature combination.

Fourth, the rendering protocol is a single global setting, using Roboto-Regular, 12 pt, and a fixed page sizing rule, applied uniformly to all benchmarks. We do not search rendering settings per dataset, and the App. H sweep is a sensitivity test, not a tuning sweep.

Fifth, foveation is a localized inverse of the push-forward map and is gated by the trigger of Eq. 16. It is not expected to repair diffuse inter-dominated cases, such as long-document summarization tasks where evidence is distributed across many rendered regions. On such inputs, the rule deliberately abstains rather than spending extra visual tokens when the cost map provides no localized target. QMSum is the main exception in our suite because its query-specific evidence is often concentrated around a small number of dialogue turns, as discussed in App. G.5.

Sixth, the headline experiments use a single backbone family, Qwen3-VL/Qwen3 at three scales. A cross-architecture replication on InternVL3.5 is reported in App. J. Macros agree to within ± 2.5 points and the per-task pattern shows ViT-family-specific differences on long-document QA and binary classification, so the framework's qualitative behavior transfers but exact gap magnitudes do not. The framework's portability beyond these two ViT families is best understood as a hypothesis derived from the push-forward map structure rather than as an experimental claim of this paper.

Seventh, the multiplicative form of the routing rule, $\mathrm{TE}(x) = (1 + \gamma - C(x)) \cdot \mathrm{VCR}(x)$, has no explicit mechanism to detect when the text path is already near its score ceiling. On near-ceiling, high-VCR external benchmarks such as the memory subtask of VTCBench, the compression factor can dominate the cost term and push $\mathrm{TE}(x)$ above the operating threshold even when the text path remains stronger. App. F.3 documents this as a structural failure mode of the present rule and as a candidate for ceiling-aware extensions in future work.

We mark these as limitations rather than weaknesses. Each scope choice is the price of obtaining a transparent, downstream-label-free framework, and each is the natural starting point for follow-up work on visual text compression. We hope that the modeling perspective and experimental design developed here can provide useful starting points for future research on visual text compression.

B Related Work on Optimal Transport and Measure-Transport Views

The purpose of this section: This section clarifies how our measure-transport formulation of VTC relates to prior uses of OT in deep learning. We do not claim novelty in OT theory, Wasserstein distances, or Sinkhorn-style solvers [35]. Our contribution is to identify the ViT visual encoder in VTC as a fixed push-forward map from rendered text units to visual tokens. We then decompose the induced distortion into precision and coverage costs, and use the resulting downstream-label-free proxy to derive a path-selection rule and a foveation mechanism. This distinction matters because much prior OT work optimizes an alignment, a pruning subset, an expert assignment, a bottleneck variable, or a model-fusion plan. Our framework instead diagnoses when an already-fixed visual encoding channel is worth using.

B.1 Main Families of OT in Deep Learning

OT for cross-modal alignment. Several works use OT to align representations across modalities. Joint Wasserstein autoencoders learn image and text embeddings with a shared latent prior and paired supervision for cross-modal retrieval and phrase localization [31]. Graph Optimal Transport formulates cross-domain alignment between image objects and text words as graph matching, using Wasserstein distance for node matching and Gromov–Wasserstein distance for structure matching as regularizers in vision-language and language tasks [6]. PLOT uses OT inside prompt learning for CLIP-like models, matching local visual features to multiple learned textual prompts with a Sinkhorn-based inner loop [4]. Cross-modal OT has also been used to transfer linguistic information from pretrained language models to CTC-based ASR acoustic encoders [27], and to improve CLIP robustness by restoring image-text alignment under perturbations at test time [54].

These works treat OT as an alignment mechanism between two representation sets. The optimized object is a cross-modal coupling, prompt alignment, training regularizer, or inference-time alignment score. Our problem is not to align image and text representations into a shared embedding space. In VTC, the text is first rendered into an image and then encoded by a fixed ViT. The key question is what task-relevant information is lost by this fixed channel. Thus, our transport object is the encoder-induced map T and the decoder-visible visual measure ν_x derived from its push-forward, rather than a learned or solved coupling between independently produced image and text representations.

OT and visual-token pruning. Visual-token pruning methods reduce the number of visual tokens processed by a multimodal model. DivPrune is a closely related non-OT baseline in this space: it selects a diverse subset of visual tokens to reduce redundancy and preserve representative visual information [1]. OTPrune makes the OT connection explicit by formulating pruning as distribution alignment: it minimizes a surrogate of the 2-Wasserstein distance between the full visual-token distribution and a pruned subset, yielding a training-free token-selection criterion [8].

This line is adjacent but structurally different from ours. Token-pruning methods operate after the visual encoder has already produced visual tokens, and their objective is to approximate the full visual-token distribution with fewer tokens. They do not decide whether the input should be read through the visual path at all. They also do not model the text-to-vision transformation that produced the visual tokens. Our rule acts before consulting downstream labels or path scores: from input-computable features and probe-calibrated parameters, it decides whether VTC is likely to preserve enough task-relevant information to justify using the visual path. Our foveation module then adds local high-resolution evidence only when the same cost model estimates that local refinement can repay its token overhead.

OT for mixture-of-experts routing. Another family uses OT for token-to-expert assignment in mixture-of-experts (MoE) models. Empirical studies of vision MoE routers compare softmax, expert-choice, token-choice, Sinkhorn, and soft-routing variants, including routers motivated by balanced assignment and OT-style constraints [26]. Selective Sinkhorn Routing formulates sparse token-to-expert assignment as an entropy-regularized OT problem and uses the transport map to promote balanced expert utilization without auxiliary load-balancing losses [33].

The shared word "routing" can be misleading here. MoE routers choose which expert processes each token inside a model forward pass. Our decision rule chooses between two input interfaces, text tokens and rendered visual tokens, before consulting downstream labels or path scores. The MoE

transport plan is the routing decision itself. Our $\mathrm{TE}(x)$ rule is a diagnostic decision derived from the estimated transport cost of a fixed visual channel. The units, timing, and objectives are therefore different: token-to-expert balancing inside a network versus input-to-path selection for VTC.

OT for information bottlenecks and model fusion. Information-bottleneck OT recasts the IB problem from a posterior-probability perspective and uses entropy-regularized OT with generalized Sinkhorn-style optimization to trace the bottleneck tradeoff [5]. Model fusion via OT aligns neurons or layer-wise units across independently trained networks before averaging their parameters, producing a post-hoc fusion plan for model merging or compression [38].

These works use OT to optimize a latent bottleneck or a model-parameter alignment. Our paper does not learn a bottleneck variable, solve for an optimal compressed representation, compute mutual information, or fuse model weights. The ViT encoder is already given. We analyze the distortion induced by that encoder when text is rendered as pixels, and use a downstream-label-free proxy $C(x)$ to decide when the resulting visual path is worth using.

B.2 What Our Formulation Adds

Across the families above, OT is usually used as an optimization tool for domain alignment, subset selection, expert-load balancing, IB-curve tracing, or model-parameter fusion. Our use is diagnostic and operational. We model an existing ViT patch encoder as a push-forward map and ask when its induced loss matters for downstream utility. Three aspects separate the proposed framework from the OT uses above.

Encoder-induced push-forward rather than learned alignment. The transformation from text to visual tokens in VTC is not an abstract cross-modal matching problem. Rendering gives each textual unit a spatial glyph support, and the ViT patch encoder aggregates these localized textual units into visual-token supports. As formalized in Sec. 3, this rendering-and-encoding pipeline can be viewed as a many-to-one map T that induces the push-forward map $T_{\#}\mu_x$. In the decoder-visible token view, this push-forward is represented by the visual-token support measure ν_x after suppressing preimage weights. This structure lets us factor the induced push-forward as

$$T_{\#} = T_{\mathrm{inter}} \circ T_{\mathrm{intra}},$$

where T_{intra} induces within-patch precision loss and T_{inter} induces cross-patch coverage fragmentation. Prior alignment works can say whether two independently produced representation sets can be matched. They do not provide this decomposition of the fixed VTC channel from rendered text units to visual tokens.

Task-aware proxy rather than task-agnostic Wasserstein distance. A raw Wasserstein distance depends on a ground metric and does not by itself encode which textual distinctions matter for a task. In VTC, the same visual compression can be harmless for coarse topic classification but damaging for candidate-list cloze QA, numerical QA, or long-document summarization. For this reason, our cost is not a numerical solution to an OT problem:

$$C(x) = \alpha W(x) + \beta L(x)\big(1 - \mathrm{TRR}(x)\big).$$

It is an operational, task-aware proxy for the labelled gap $\Delta(x) = s_{\mathrm{text}}(x) - s_{\mathrm{vis}}(x)$. Its form is motivated by the push-forward factorization, while the scalars (α, β, γ) are estimated from synthetic downstream-label-free probes rather than from downstream benchmark labels.

Path-level decision and patch-level repair share one cost model. The same probe-derived cost parameters underlie two decisions. Globally, $\mathrm{TE}(x) = (1 + \gamma - C(x))\mathrm{VCR}(x)$ selects whether to use the visual path. Locally, the patch cost C_q identifies high-cost regions for foveation when the trigger in Sec. 3.3 estimates that extra visual tokens can recover enough information. This shared structure is different from a pipeline that first prunes tokens by one criterion and later routes by another. It is also different from MoE routing, where the transport plan directly assigns tokens to experts but does not estimate whether an alternative input channel should be used.

B.3 Adjacent VTC Critiques

Two recent critiques of visual text interfaces help define the boundary of our claim. One recent study [22] argues that rendering subword embeddings as pixels and re-encoding them can behave like an inefficient autoencoder, and that simple pooling can be competitive on token-level reconstruction objectives. This does not contradict our claim, because our objective is not token reconstruction. The framework predicts that the visual path should be avoided when the rendered channel destroys task-relevant distinctions, and used only when preserved layout or evidence locality outweighs transport cost. The structured-format bonus γ and the routing rule make this distinction explicit rather than assuming that visual compression is always useful.

Work on DeepSeek-OCR's linguistic-crutch behavior shows that high OCR or reconstruction quality does not guarantee reliable reasoning from visual text [24]. Our precision term $\alpha W(x)$ targets this type of failure mode. Tasks whose answers require lexical, numerical, or candidate-level fidelity receive higher precision cost and are more likely to be routed to the text path. Thus, the framework does not deny these failures. It turns them into a measurable cost that governs path selection.

B.4 What Is and Is Not New

We do not claim novelty in OT itself, in Wasserstein distances as similarity measures, or in Sinkhorn-style optimization. We also do not claim to compute a true Wasserstein distance between text and visual-token measures for each input. The novelty is the use of the measure-transport viewpoint to identify the right diagnostic object for VTC and to turn that object into operational decisions. Concretely, the paper contributes:

- A push-forward view of ViT-based VTC, with the encoder factorized into within-patch aggregation and cross-patch arrangement.
- A downstream-label-free proxy $C(x)$ whose precision and coverage terms are separately estimable and whose ordering is validated against paired LLM/VLM behavior.
- A path-selection rule that trades transport cost against visual-token savings and decides whether to use the visual path without downstream-label calibration.
- A foveation mechanism that reuses the same cost parameters at the patch level, so local repair is tied to the same transport-cost structure as global routing.

This is why the closest OT literature should be read as methodological context rather than as prior versions of the same contribution. Existing OT works optimize alignments, subsets, expert assignments, bottlenecks, or model parameters. Our work instead diagnoses the task-relevant loss of a fixed visual text compression channel, uses this diagnosis to decide when visual compression is worth using, and applies the same cost structure to repair localized visual-token loss.

In short, the contribution is not a new OT solver, but a transport-based diagnostic framework for deciding when and where visual text compression should be trusted.

C Benchmark Suite and Evaluation Protocol

The purpose of this section: This section makes the empirical evidence and the ***downstream-label-free claim*** reproducible by documenting the full evaluation suite, input statistics, answer formats used to assign $W(x)$, and the prompt protocol used for all LLM/VLM comparisons.

C.1 Benchmark Suite

Table 5: The 24-benchmark evaluation suite. ***Samples*** is the number of evaluation instances used in every run reported here. ***Avg. tokens*** is the mean LLM-path input length per sample, computed with the Qwen3 tokenizer and including the prompt template. ***Answer format*** is the output constraint enforced by the instruction and provides the basis for assigning the precision feature $W(x)$ in Eq. 8. Metric abbreviations: Acc. = accuracy, R1 = Rouge-1. Datasets are grouped by broad task family.

Dataset	Task family	Metric	Samples	Avg. tokens	Answer format
RACE-H [21]	Multi-choice RC	Acc.	3,451	469	One letter A / B / C / D
QuALITY [34]	Multi-choice RC	Acc.	2,086	5,039	One letter A / B / C / D
LogiQA [25]	Multi-choice RC	Acc.	651	237	One letter A / B / C / D
DREAM [39]	Multi-choice RC	Acc.	2,041	235	One letter A / B / C
BoolQ [10]	Binary QA	Acc.	3,270	189	Yes / No
IMDB [30]	Binary sentiment	Acc.	25,000	334	Positive / Negative
DBpedia-14 [51]	14-way topic	Acc.	70,000	148	One of 14 category names
Yahoo Answers [51]	10-way topic	Acc.	60,000	200	One of 10 category names
Yelp Full [51]	5-way sentiment	Acc.	50,000	253	Integer 1–5
ContractNLI [19]	3-way NLI	Acc.	1,037	2,370	Entailment / Contradiction / N.M.
SciFact [43]	Claim verification	F1	300	595	Rationale + SUPPORT / CONTRADICT
HotpotQA [49]	Multi-hop QA	F1	7,405	1,516	Free-form short
MuSiQue [41]	Multi-hop QA	F1	2,417	2,584	Free-form short
2WikiMultiHopQA [16]	Multi-hop QA	F1	12,576	1,070	Free-form short
CoQA [36]	Conversational QA	F1	7,983	404	Free-form short
DROP [13]	Numerical QA	F1	9,535	344	Number / span / date
MultiFieldQA-en [2]	Long-doc. QA	F1	150	4,905	Free-form short
QASPER [11]	Scientific QA	F1	1,451	4,228	Yes / No / Unanswerable / free-form
ReCoRD [50]	Cloze QA	F1	10,000	341	Entity from candidate list
CNN/DailyMail [15]	News summ.	R1	11,490	732	Free-form summary
XSum [32]	News summ.	R1	11,334	519	Single-sentence summary
BillSum [20]	Legal summ.	R1	3,269	2,080	Free-form summary
GovReport [17]	Long-doc. summ.	R1	972	9,177	Executive summary
QMSum [53]	Query-based summ.	R1	272	12,241	Query-specific summary

The purpose of this subsection: This subsection specifies 24 evaluation datasets and explains why the suite is suitable for testing our cost model: it spans ***precision-sensitive*** tasks, where $W(x)$ should dominate, and ***coverage-sensitive*** tasks, where $L(x)(1 - \mathrm{TRR}(x))$ should dominate.

We evaluate on 24 established English NLP benchmarks spanning six broad task families: (i) multi-choice reading comprehension, (ii) binary classification, (iii) multi-class classification, (iv) natural-language inference and claim verification, (v) short-answer question answering, and (vi) long-document summarization. The suite covers tasks with short, label-structured outputs, such as single letters, category names, and Yes/No answers, and longer free-form outputs, such as multi-sentence summaries and span-style answers. This diversity is important for our cost model: the precision feature $W(x)$ in Eq. 8 depends on answer format, while the coverage feature $L(x)$ depends on how widely task-relevant evidence is distributed across the input. Tab. 5 summarizes the task family, metric, sample count, average input length, and answer format for each dataset.

C.2 Prompt Parity between the Text and Visual Paths

Table 6: LLM and VLM instruction templates using RACE-H [21] as an example. The two prompts differ only in the italicized surface pointer to the input channel. The task description and answer-format constraint are otherwise identical.

LLM	*Read the following passage* and question, then choose the correct answer from A, B, C, or D. Reply with only one letter: A, B, C, or D.
VLM	*Read the passage shown in the image* and the following question, then choose the correct answer from A, B, C, or D. Reply with only one letter: A, B, C, or D.

The purpose of this subsection: This subsection establishes that the LLM and VLM paths use ***matched task semantics***, so the measured gap $\Delta(x) = s_{\text{text}}(x) - s_{\text{vis}}(x)$ can be interpreted as an operational estimate of visual-encoding cost rather than a prompt artifact.

Every LLM/VLM comparison in this paper uses textually parallel and semantically matched prompts. Both paths receive the same task instruction, the same question, and the same passage content. The controlled distinction is the channel through which the passage enters the model. For each task family, we maintain one instruction template whose LLM and VLM variants differ only in a surface pointer to the input channel: the LLM variant introduces the passage with ***"Read the following passage"***, while the VLM variant introduces the identical passage, rendered as an image, with ***"Read the passage shown in the image"***. An example for the RACE-H benchmark appears in Tab. 6. Three properties follow from this protocol and jointly support the fairness of comparisons.

(1) Task semantics are matched. The instruction that defines the task, including what question to answer and what answer format to use, is identical up to the channel pointer. Neither path receives extra guidance, label hints, or chain-of-thought scaffolding.

(2) The underlying passage content is held fixed. The VLM image is rendered from exactly the same passage that the LLM receives as text. Thus, before model-specific encoding, the two paths are given the same underlying content. Any performance difference is not caused by changing the passage, adding evidence, or removing evidence, but by changing the representation channel through which the passage is consumed.

(3) The controlled distinction is the visual encoding pipeline. Because the task instruction, question, answer constraint, and underlying passage content are held fixed, the relevant distinction between $s_{\text{text}}(x)$ and $s_{\text{vis}}(x)$ is whether the passage is consumed directly as subword tokens or first rendered and encoded through the visual pipeline modeled as the ViT push-forward map in Sec. 3. We therefore interpret the empirical gap $\Delta(x) = s_{\text{text}}(x) - s_{\text{vis}}(x)$ as an operational measure of the cost introduced by the ***visual encoding pipeline***, which is the quantity that the proxy $C(x)$ is designed to estimate.

Table 7: Representative LLM-path and VLM-path instruction templates for the six broad task families. For each family, the LLM variant introduces the passage with "*Read the following* $\langle X \rangle$", while the VLM variant introduces the identical passage, rendered as an image, with "*Read the* $\langle X \rangle$ *shown in the image*". The remaining task description and answer-format constraint are otherwise matched across the two paths. The precision feature $W(x)$ in Eq. 8 reads directly from the ***Answer format*** column.

Task family	Path	Representative instruction	Answer format
Multi-choice RC	LLM	*Read the following passage and question, then choose the correct answer from A, B, C, or D. Reply with only one letter: A, B, C, or D.*	One letter
	VLM	*Read the passage shown in the image and the following question, then choose the correct answer from A, B, C, or D. Reply with only one letter: A, B, C, or D.*	
Yes/No QA	LLM	*Read the following passage and answer the question with Yes or No. Reply with exactly one word: Yes or No.*	One word
	VLM	*Read the passage shown in the image and answer the question with Yes or No. Reply with exactly one word: Yes or No.*	
Multi-class	LLM	*Classify the following text into one of these 14 categories: Company, . . . , Written Work. Reply with exactly one category name.*	One category name
	VLM	*Classify the text shown in the image into one of these 14 categories: Company, . . . , Written Work. Reply with exactly one category name.*	
NLI / verif.	LLM	*Given a contract and a hypothesis about the contract, determine whether the hypothesis is Entailment, Contradiction, or Not mentioned. Reply with exactly one of: Entailment, Contradiction, or Not mentioned.*	One of 3 labels
	VLM	*Given the contract shown in the image and a hypothesis about the contract, determine whether the hypothesis is Entailment, Contradiction, or Not mentioned. Reply with exactly one of: Entailment, Contradiction, or Not mentioned.*	
Short-answer QA	LLM	*Read the following paragraphs and answer the question. The question may require combining information from multiple paragraphs. Give a concise, direct answer.*	Free-form short
	VLM	*Read the paragraphs shown in the image and answer the question. The question may require combining information from multiple paragraphs. Give a concise, direct answer.*	
Summarization	LLM	*Summarize the following text concisely. Write a summary that captures the key points.*	Free-form summary
	VLM	*Summarize the text shown in the image concisely. Write a summary that captures the key points.*	

C.3 Prompt Design Rationale

The purpose of this subsection: This subsection explains how the prompts are constructed without downstream-label tuning, and why the answer-format column in Tab. 5 provides a reproducible basis for assigning the precision feature $W(x)$.

For each task, we follow two design rules. First, the ***label space*** is inherited directly from the dataset's official schema: for example, DREAM uses A/B/C because the dataset has three options, DBpedia lists all 14 official category names, and Yelp Full uses the 1–5 rating scale. Second, the ***instruction phrasing*** follows common lm-evaluation-harness [3] and LongBench [2] conventions for single-token classification outputs and concise free-form QA outputs, respectively. These rules ensure that the label-free precision feature $W(x)$ in Eq. 8 maps cleanly from the answer-format column of Tab. 5 to a task-sensitivity score, without introducing prompt-engineering degrees of freedom tuned on downstream labels. One LLM/VLM instruction pair per family is shown in Tab. 7, confirming that the channel-pointer substitution of Sec. C.2 is applied uniformly across all six families. The remaining datasets follow the same within-family templates, and their full instruction text and output parsers will be released with the code.

D Probe-Based Calibration of (α, β, γ) and Sample-Paired Validation

The purpose of this section: This section makes the calibration of the transport-cost model traceable and downstream-label-free. The decomposition $C(x) = \alpha W(x) + \beta L(x)(1 - \mathrm{TRR}(x))$ depends on calibrated scalars (α, β, γ) whose values are used in the main text but not derived there. We first describe the three synthetic probes used to estimate these scalars at each backbone scale. The probes use synthetic gold answers only inside the probe tasks and never use labels, scores, or outcomes from the 24 downstream benchmarks. We then perform a separate post-hoc validation to test whether the resulting fixed $C(x)$ predicts paired VLM-vs-LLM behavior on real evaluation samples. This second analysis is ***validation, not calibration***: it does not modify $C(x)$, (α, β, γ), the routing threshold τ, or any foveation decision. Its role is only to address the concern that $C(x)$ might be a pure task-family lookup rather than a label-free transport-cost ordering.

D.1 Synthetic Probes for (α, β, γ)

The purpose of this subsection: This subsection defines the three synthetic probe datasets, the tier structure of each probe, the metric used per tier, and the regression fit that converts per-tier VLM-LLM gaps into one scalar. All three probes use information-dense synthetic content, paired VLM-vs-LLM evaluation under matched prompts, and per-tier accuracy aggregated into one parameter through a no-intercept least-squares fit. These probes are the only source of (α, β, γ).

α probe, generation fidelity. The α probe measures how the precision of VLM output decays as the answer becomes more specific. We construct 80 information-dense needles, each a short biographical fact about a person, university, year, and award. The model performs one of three output tasks with increasing precision demand: extract a single keyword (Low, $W = 0.10$), produce a one-sentence summary (Medium, $W = 0.35$), or restate the full needle in roughly 50 words (High, $W = 0.65$). Each needle is rendered into both a 1,000-character and a 4,000-character haystack, giving $80 \times 3 \times 2 = 480$ trials per model. Low-tier responses are scored by exact-substring match, and the medium and high tiers by ROUGE-L F1 against the gold needle. Per tier we compute the gap

$$g_W = \max\bigl(0, 1 - \mathrm{VLM}_{\mathrm{avg}}/\mathrm{LLM}_{\mathrm{avg}}\bigr),$$

then fit

$$\alpha = \frac{\sum_W W \cdot g_W}{\sum_W W^2}, \qquad R^2 = 1 - \frac{\sum_W (g_W - \alpha W)^2}{\sum_W g_W^2}.$$

This is a no-intercept least-squares fit of g_W on W over tier means.

β probe, multi-fact aggregation. The β probe measures how VLM performance degrades when the answer requires aggregating evidence dispersed across the rendered page. We generate 60 city-temperature aggregation tasks, such as asking which listed city had the highest temperature. Each task has three locality tiers: a single fact placed in one location (Low, $L = 0.10$), three facts scattered across the page (Medium, $L = 0.40$), and five facts placed at 10%, 30%, 50%, 70%, and 90% of the document (High, $L = 0.70$). Each task is rendered in two haystack lengths, giving $60 \times 3 \times 2 = 360$ trials per model. Numeric answers are scored with a $\pm 15\%$ tolerance and name answers by exact match. The fit is identical to the α case but with L replacing W:

$$\beta = \frac{\sum_L L \cdot g_L}{\sum_L L^2}, \qquad R^2 = 1 - \frac{\sum_L (g_L - \beta L)^2}{\sum_L g_L^2}.$$

γ probe, structured-format bonus. The γ probe isolates the visual encoder's ability to exploit structured layout. We generate 60 tasks where the same set of facts about three people, their award years, and their award names is presented in two formats: a Markdown table (*structured*) and a plain paragraph (*flat*). The model is asked who received their award most recently, scored by name-substring match. Across two haystack lengths this gives $60 \times 2 \times 2 = 240$ trials per model. We compute the per-format VLM-LLM accuracy ratios

$$r_{\mathrm{struct}} = \mathrm{VLM}_{\mathrm{struct}}/\mathrm{LLM}_{\mathrm{struct}}, \qquad r_{\mathrm{flat}} = \mathrm{VLM}_{\mathrm{flat}}/\mathrm{LLM}_{\mathrm{flat}},$$

and define the structured bonus as

$$\gamma = \max\bigl(0,\ r_{\mathrm{struct}} - r_{\mathrm{flat}}\bigr).$$

Unlike α and β, γ has no R^2 because there are only two format conditions.

Table 8: Per-tier VLM/LLM accuracies and fitted (α, β) across model scales. "Low/Med/High" are the three tier values of W for the α probe or L for the β probe. The gap g is $\max(0,\, 1 - \mathrm{VLM}_{\mathrm{avg}}/\mathrm{LLM}_{\mathrm{avg}})$. Each tier aggregates 160 trials per model for the α probe and 120 for the β probe. The fitted parameter is a no-intercept least-squares fit of g on the tier value, with R^2 measuring the fraction of g^2 explained by the linear fit.

	Low (tier = 0.10)		Medium (0.35 / 0.40)		High (0.65 / 0.70)		Fit	
Probe @ scale	VLM	LLM	VLM	LLM	VLM	LLM	param	R^2
α @ 4B	0.794	1.000	0.531	0.642	0.651	0.690	0.213	0.33
α @ 8B	0.906	1.000	0.526	0.632	0.547	0.764	0.455	0.98
α @ 32B	0.944	1.000	0.628	0.566	0.843	0.875	**0.053**	0.35
β @ 4B	0.975	1.000	0.650	0.883	0.442	0.783	0.627	0.99
β @ 8B	0.992	1.000	0.800	0.850	0.717	0.733	**0.061**	0.61
β @ 32B	0.992	1.000	0.825	0.933	0.742	0.875	0.233	0.98

Table 9: γ probe per-format ratios and the fitted (α, β, γ) triplet used in $C(x)$ and $\mathrm{TE}(x)$. The ratio $r = \mathrm{VLM}/\mathrm{LLM}$ is computed separately on the structured Markdown-table condition and the flat paragraph condition. The γ value is the non-negative excess ratio. The right block restates the calibrated triplet at each scale.

	γ probe ratios			Calibrated (α, β, γ)			
Scale	r_{struct}	r_{flat}	γ	α	β	γ	Cost-map regime
4B	0.961	0.892	0.069	0.213	0.627	0.069	both terms active
8B	0.887	0.889	0.000	0.455	**0.061**	0.000	precision-dominated
32B	1.031	0.790	0.241	**0.053**	0.233	0.241	coverage-dominated

Per-tier accuracies and fitted parameters. Tab. 8 reports the per-tier VLM and LLM accuracies and the resulting gap for the α and β probes at all three scales, together with the fitted α, β, and the corresponding R^2. Tab. 9 reports the per-format ratios for the γ probe and the final (α, β, γ) triplet used in $C(x)$ and $\mathrm{TE}(x)$ throughout the rest of the paper. The same calibrated values are used by the scale-dependent foveation analysis in App. G.

Two patterns in Tab. 8 and Tab. 9 are worth noting because they motivate App. G. At 8B, the β probe yields $\beta = 0.061$ with $R^2 = 0.61$. The per-tier gaps are weak and non-monotonic (0.008, 0.059, 0.023), so the linear fit is small and only weakly supported by the data. At 32B, the α probe yields $\alpha = 0.053$ with $R^2 = 0.35$. The medium-tier VLM accuracy exceeds the LLM accuracy (0.628 vs. 0.566), so the corresponding gap collapses to zero and the linear fit explains little residual variance. We use these probe-level R^2 values as evidence that the cost decomposition becomes scale-dependent. That same scale dependence drives the foveation behavior analyzed in App. G.

D.2 Sample-Paired Validation of $C(x)$

Table 10: Mean per-sample visual advantage $A(x) = s_{\mathrm{VLM}}(x) - s_{\mathrm{LLM}}(x)$ within each $C(x)$ quantile bucket on 4B suite. The analysis uses approximately 230,000 paired sub-samples drawn from the 4B evaluation outputs. Win rate is the fraction of samples in the bucket where VLM strictly beats LLM.

Bucket	$C(x)$ range	n	Mean A	Win rate	Task families
1	$[0.085, 0.181)$	18,680	$+0.170$	39.3%	long-context QA + multi-choice RC
2	$[0.181, 0.186)$	70,000	$+0.052$	9.3%	topic classification
3	$[0.186, 0.191)$	61,037	$+0.095$	17.7%	topic classification + NLI
4	$[0.191, 0.639]$	79,948	-0.085	20.5%	short-context classification + QA

Table 11: Mean visual advantage $A(x)$ within a 3×3 joint partition of samples by $\mathrm{VCR}(x)$ tertile (rows) and $C(x)$ tertile (columns). High-C cells have the lowest mean visual advantage within each VCR row, showing that $C(x)$ predicts VLM-vs-LLM behavior beyond compression ratio alone. Cell counts are shown in parentheses.

	Low C	Mid C	High C
Low VCR	$+0.252$ ($n = 6{,}785$)	$+0.067$ ($n = 45{,}696$)	-0.097 ($n = 22{,}529$)
Mid VCR	$+0.153$ ($n = 8{,}608$)	$+0.075$ ($n = 50{,}876$)	-0.011 ($n = 18{,}853$)
High VCR	$+0.046$ ($n = 3{,}287$)	$+0.074$ ($n = 34{,}465$)	-0.113 ($n = 38{,}566$)

The purpose of this subsection: This subsection tests whether the probe-calibrated cost $C(x)$ predicts real benchmark behavior without fitting any downstream labels. The analysis is sample-paired because it compares VLM and LLM scores on the same evaluation samples. At the same time, the current implementation assigns one $C(x)$ value per dataset, since W is task-level and (L, TRR) are summarized at the dataset level. The goal is therefore to test whether this ***label-free dataset-level cost ordering*** remains predictive over many paired samples, rather than to claim that within-dataset sample variation is fully captured. Crucially, this subsection is not used to calibrate the method. It is a post-hoc validation of a cost function that has already been fixed by the synthetic probes.

Setup. We use 4B calibrated parameters $(\alpha, \beta) = (0.213, 0.627)$ from Tab. 9. These parameters are fixed before this analysis and are estimated only from synthetic probes, not from the 24 downstream benchmarks. We then collect aligned per-sample LLM and VLM scores from the 4B evaluation outputs solely as a ***post-hoc validation target*** for the cost ordering. This yields approximately 230,000 paired samples. For each paired sample, we assign $C(x)$ from the dataset's (W, L, TRR) triplet using the same formula as in the main method. This assignment uses only task and input-side features, and never uses model scores or ground-truth performance outcomes. No downstream score, including the visual advantage $A(x)$, is used to fit $C(x)$, α, β, γ, the routing threshold τ, or any foveation decision. Equivalently, removing this validation analysis would not change any calibrated parameter, routing choice, foveation choice, or main result. We compute the per-sample visual advantage

$$A(x) = s_{\mathrm{VLM}}(x) - s_{\mathrm{LLM}}(x).$$

Positive $A(x)$ means the visual path outperforms the text path, while negative $A(x)$ means the text path outperforms the visual path.

Spearman correlation. The Spearman rank correlation between $C(x)$ and $A(x)$ across all of these paired samples is

$$\rho\big(C(x), A(x)\big) = -0.125, \qquad p < 10^{-300}.$$

The negative sign matches the qualitative claim: higher-cost datasets have systematically lower visual advantage. Thus, this test should be read as a ***sample-paired validation of cost ordering***, not as evidence that the current proxy captures all within-dataset variation.

Quantile bucketing on $C(x)$. We partition the samples into four quantile buckets by $C(x)$. The four buckets are reported in Tab. 10. The lowest-C bucket has the largest mean visual advantage, $+0.170$, while the highest-C bucket is the only bucket with negative mean visual advantage, -0.085. The middle buckets remain positive and close in magnitude, which reflects the coarse dataset-level nature of $C(x)$ in this validation.

Controlling for the visual compression ratio. A reviewer might worry that the negative correlation in Tab. 10 is driven entirely by the visual compression ratio $\mathrm{VCR}(x)$, since samples with different compression ratios may also have different task profiles. To rule this out, we partition these paired samples into a 3×3 grid: tertiles of $\mathrm{VCR}(x)$ along the rows and tertiles of $C(x)$ along the columns. Tab. 11 reports the within-cell mean visual advantage $A(x)$. Within every VCR row, the high-C cell has the lowest mean visual advantage: -0.097 in the low-VCR row, -0.011 in the mid-VCR row, and -0.113 in the high-VCR row. Thus, $C(x)$ contributes information about visual advantage that is not absorbed by compression ratio.

Together, the probe-derived calibration in App. D.1 and the sample-paired validation in Tab. 10 and Tab. 11 make the cost decomposition traceable end to end. The parameters come from independent synthetic probes with measured R^2 values, and the resulting $C(x)$ predicts real-data visual advantage over paired samples even after controlling for visual compression ratio. This validation does not participate in calibration. It only tests whether the already-fixed cost ordering has the expected relationship with downstream behavior. The decision-rule sensitivity in App. F and the foveation limitations in App. G both use the probe calibration defined in this section.

E Per-Dataset Cross-Scale Performance

The purpose of this section: This section tests whether the transport-cost explanation holds beyond the macro-averaged 4B result in the main paper. It reports cross-scale results at Qwen3-4B, Qwen3-8B, and Qwen3-32B and asks whether the qualitative LLM/VLM preference predicted by $C(x)$ persists at the task-family and per-dataset levels. The main paper reports macro-averaged Qwen3-4B results in Tab. 2. We first aggregate the per-dataset results by task family and compare the observed LLM/VLM gaps with the regimes predicted by the cost decomposition in Eq. 8. The full per-dataset grid is provided in the following subsection.

E.1 Theory–Data Validation by Task Family

Table 12: Per-family theory-data validation. Predicted regimes are derived from the dominant term of $C(x)$ in Eq. 8. Pred. sign denotes the expected LLM/VLM gap direction. "+" means LLM wins, "−" means VLM is non-degraded or better, and "N/A" means no fixed prediction for the mixed regime. The Δ columns report observed family-average gaps at Qwen3-4B, Qwen3-8B, and Qwen3-32B. Positive values mean LLM beats VLM. Family Δ values are computed from full-precision LLM and VLM scores and rounded to two decimals for display. Recomputing Δ from the 1dp scores in Tab. 13 can give values differing by up to 0.06 across the 18 cells of this table due to per-dataset rounding accumulation.

Task family	Predicted regime	Pred. sign	4B Δ	8B Δ	32B Δ	Match?
Multi-choice RC	intra ($\uparrow W$)	$+$	$+9.07$	$+2.65$	$+8.28$	✓
Binary	intra ($\uparrow W$)	$+$	$+9.64$	$+5.38$	$+5.29$	✓
Multi-class	low cost	$-$	-2.45	-3.74	-0.33	✓
NLI/verif.	mixed	N/A	-5.26	$+2.13$	$+4.42$	N/A
Short-answer QA	mixed	N/A	-2.98	-0.02	$+1.89$	N/A
Summarization	inter ($\uparrow L$)	$+$	$+4.73$	$+2.25$	$+1.69$	✓
All ($n=24$)	N/A	N/A	$+1.56$	$+1.07$	$+3.13$	N/A

The purpose of this subsection: This subsection validates the ***explanatory content*** of the cost decomposition rather than the final routing rule. Specifically, it asks whether the dominant term in $C(x)$ predicts the sign of the LLM/VLM gap at the task-family level.

The cost decomposition

$$C(x) = \alpha W(x) + \beta L(x)(1 - \mathrm{TRR}(x)) \tag{17}$$

implies different qualitative regimes across task families. When $W(x)$ is high and $L(x)$ is small, the dominant loss is *intra-patch* lexical smoothing, so the text path is expected to outperform the visual path. When $L(x)$ is high, the dominant loss is *inter-patch* fragmentation, so the text path is again expected to be favored. When both terms are small, the visual path should remain non-degraded or better, because the induced transport cost is low enough that the visual representation remains usable. Finally, when the two terms are comparable, the family falls into a *mixed* regime, where the sign of the LLM/VLM gap is not expected to be fixed and may depend on model scale.

Tab. 12 reports the predicted regime, the corresponding expected sign of the LLM/VLM gap, and the observed family-average gap $\Delta_{\mathrm{LLM-VLM}} = s_{\mathrm{text}} - s_{\mathrm{vis}}$ at all three scales. Positive $\Delta_{\mathrm{LLM-VLM}}$ means that the LLM path outperforms the VLM path, while negative values mean that the VLM path performs better. For mixed regimes, we mark the predicted sign as N/A because the cost decomposition does not imply a fixed direction.

The observed signs align with the model's directional expectations for all four families where the theory makes a fixed prediction. *Multi-choice RC* and *Binary* fall into the *intra* regime. Their outputs require exact option-level or label-level distinctions, making them sensitive to within-patch lexical smoothing. Accordingly, the LLM path consistently outperforms the VLM path, with 4B gaps of $+9.07$ and $+9.64$ points. *Summarization* falls into the *inter* regime. Task-relevant evidence is distributed across long inputs, making cross-patch fragmentation costly. This yields an LLM advantage of $+4.73$ points at 4B, with the same sign preserved at 8B and 32B. In contrast, *Multi-class* classification falls into the *low-cost* regime. Inputs are short, labels are coarse, and local lexical

smoothing rarely changes the required category. The VLM path therefore remains non-degraded and slightly outperforms the LLM path across all three scales.

The two families identified as *mixed*, *NLI/verif.* and *Short-answer QA*, show bidirectional behavior as expected. At 4B, the visual path wins by -5.26 and -2.98 points, respectively, but the gaps shrink or flip at larger scales. This scale-dependent behavior is consistent with both components of $C(x)$ being non-negligible. Neither intra-patch smoothing nor inter-patch fragmentation alone dominates the family, so the final LLM/VLM preference depends on the backbone's ability to recover information from the visual representation.

E.2 Full Per-Dataset Breakdown

Table 13: Per-dataset performance at the 4B, 8B, and 32B scales for the three inference strategies considered in Sec. 4: text-only LLM, text-rendered-as-image VLM, and VLM with localized inverse transport (Foveation, FOV). Within each dataset and scale, the best score among the three methods is boldfaced. Macro rows aggregate over S_2/S_3 (the 4B visual-friendly partition held fixed across all three scale columns; per-scale partitions are reported separately in App. J) and over all 24 datasets.

	4B			8B			32B		
Dataset	LLM	VLM	FOV	LLM	VLM	FOV	LLM	VLM	FOV
Multi-choice RC									
RACE-H	25.7	25.1	**25.9**	**25.0**	23.9	24.8	25.6	**26.7**	25.7
QuALITY	**67.8**	60.5	60.5	**70.3**	67.2	68.2	**81.2**	75.3	75.0
LogiQA	**50.2**	38.9	39.3	**43.2**	41.3	41.3	**61.6**	42.9	42.9
DREAM	**88.0**	71.1	71.1	**86.4**	81.9	81.5	**95.1**	85.4	85.9
Binary									
BoolQ	**85.4**	78.3	78.3	**86.9**	84.2	83.8	**89.0**	87.2	87.6
IMDB	**94.6**	82.4	82.4	**94.7**	86.7	86.7	**94.6**	85.7	85.7
Multi-class									
DBpedia	88.7	**93.8**	**93.8**	92.2	**96.5**	96.5	95.1	**97.3**	97.3
Yahoo	54.3	64.1	**64.3**	59.1	**66.7**	66.6	66.5	**67.7**	67.6
Yelp	**53.4**	45.8	45.8	**53.9**	53.2	53.2	**57.6**	55.3	55.3
NLI/verif.									
ContractNLI	**51.5**	41.9	42.6	**53.3**	41.1	40.3	**57.8**	41.3	39.7
SciFact	51.3	71.4	**72.2**	62.4	**70.4**	70.4	67.3	**74.9**	74.9
Short-answer QA									
HotpotQA	13.1	35.4	**36.4**	26.1	**37.7**	36.1	17.8	29.3	**33.6**
MuSiQue	8.0	19.0	**20.1**	16.3	**22.8**	21.8	11.4	18.9	**21.4**
2WikiMHQA	13.4	38.2	**38.5**	26.6	**44.0**	40.7	19.3	35.7	**39.5**
CoQA	**41.5**	39.2	39.3	**52.3**	48.6	48.8	**41.6**	25.3	26.0
DROP	18.5	**22.0**	**22.0**	**35.6**	26.2	26.4	**23.9**	21.1	21.3
MultiFieldQA	**46.8**	38.8	39.2	**47.0**	44.1	44.4	42.6	41.7	**43.0**
QASPER	32.7	30.1	**33.0**	34.7	**45.0**	44.4	32.2	31.7	**32.4**
ReCoRD	**51.3**	26.5	27.0	**58.7**	29.0	28.5	**76.8**	46.8	46.8
Summarization									
CNN/DM	**34.0**	24.1	24.1	**34.6**	26.8	26.8	**34.3**	28.2	28.2
XSum	**18.1**	16.8	16.8	**19.0**	17.4	17.4	**19.1**	17.7	17.7
BillSum	40.9	**40.9**	40.9	38.2	**41.2**	41.2	41.7	**42.3**	42.3
GovReport	**42.0**	32.5	32.5	**39.8**	37.1	37.1	**43.6**	41.8	41.8
QMSum	**30.7**	28.0	28.8	**31.0**	29.0	28.9	29.8	30.1	**30.4**
Macro averages									
S_2 at 4B ($n = 10$)	34.7	44.0	44.7	41.6	47.4	46.9	40.1	44.6	45.6
S_3 at 4B ($n = 14$)	54.0	44.6	44.8	55.1	49.1	49.1	58.9	50.3	50.4
All ($n = 24$)	45.9	44.4	44.8	49.5	48.4	48.2	51.1	47.9	48.4

The purpose of this subsection: This subsection serves three purposes. First, it exposes the per-dataset outcomes behind the macro-averages reported in the main text, showing that the aggregate trends are not driven by a small number of outliers. Second, it tests whether the same text-friendly and visual-friendly regimes persist across model scales. Third, it separates the role of foveation from routing. FOV tests whether localized inverse transport can repair high-cost visual regions, while the final path decision is still handled by the transport-efficiency rule.

Tab. 13 reports the per-dataset scores for all 24 benchmarks, grouped by task family. Within each dataset and scale, the best score among LLM, VLM, and FOV is boldfaced, making the text-friendly and visual-friendly regimes directly visible.

Four cross-scale patterns are worth highlighting.

Intra-dominated, text-favored. Intra-sensitive families remain text-favored at every scale. At 4B, DREAM (88.0 vs 71.1), QuALITY (67.8 vs 60.5), and LogiQA (50.2 vs 38.9) in *Multi-choice RC*, BoolQ (85.4 vs 78.3) and IMDB (94.6 vs 82.4) in *Binary*, ContractNLI (51.5 vs 41.9) in *NLI/verif.*, and ReCoRD (51.3 vs 26.5) in *Short-answer QA* all show LLM path beating VLM path by at least 5 points. The same sign is preserved at 8B/32B for every dataset listed above. This is consistent with the precision-cost term in Eq. 8: outputs that depend on exact option-level, label-level, or span-level distinctions are vulnerable to within-patch lexical smoothing, regardless of backbone capacity.

Low-cost or visually recoverable, visual-friendly. The visual path is most competitive on tasks whose answer formats tolerate local lexical smoothing or whose evidence remains visually recoverable. At 4B, DBpedia (93.8 vs 88.7) and Yahoo (64.1 vs 54.3) in *Multi-class*, SciFact (71.4 vs 51.3) in *NLI/verif.*, and HotpotQA (35.4 vs 13.1), MuSiQue (19.0 vs 8.0), and 2WikiMultiHopQA (38.2 vs 13.4) in *Short-answer QA* all show the VLM path beating the LLM path. These cases are precisely those where rendering preserves the page-level locality structure used by the visual path, while the answer format does not require exact intra-patch lexical fidelity.

Inter-dominated summarization, text-favored. Long-document summarization remains text-favored even at larger scales. At 4B, CNN/DM (34.0 vs 24.1), XSum (18.1 vs 16.8), and GovReport (42.0 vs 32.5) favor the LLM path. The sign is preserved at 8B and 32B. This reflects the coverage-cost term: summarization requires integrating evidence distributed across many rendered regions, so cross-patch fragmentation accumulates over the full input. The exception is BillSum, where the VLM path is competitive at 8B and 32B, consistent with its more templated and locally compressible structure.

Foveation as bounded repair. Foveation provides a complementary but bounded repair mechanism. It improves the 4B macro-average visual score from 44.4 to 44.8 and the 32B macro-average visual score from 47.9 to 48.4, with the largest gains concentrated in tasks where a small number of high-cost regions carry useful evidence: HotpotQA (+1.0), MuSiQue (+1.1), 2WikiMultiHopQA (+0.3), QASPER (+2.9), and QMSum (+0.8) at 4B. At the same time, FOV does not uniformly dominate VLM, especially at 8B, confirming the role of the trigger in Sec. 3.3. Localized inverse transport is useful when recoverable local cost exceeds the additional token overhead, but it cannot replace the global path-selection decision.

F Decision-Rule Sensitivity and Cross-Scale Performance

The purpose of this section: This section tests whether the transport-efficiency framework remains stable under threshold changes and model scaling. The main paper reports the Qwen3-4B operating point used for the primary claim. Here we provide post-hoc threshold sweeps and cross-scale diagnostics at Qwen3-4B, Qwen3-8B, and Qwen3-32B. The analysis is intended to diagnose robustness and failure modes, not to refit the main Qwen3-4B decision rule.

F.1 Threshold Sensitivity and Cross-Scale Diagnostics

Table 14: Decision-rule accuracy versus threshold τ across model scales. The 4B column reports the standard TE rule used for the main result. The 32B column uses the bounded-benefit variant defined in App. F.2. Bolded rows mark per-scale diagnostic peaks. The sweep shows that $\tau = 1.28$ is stable for the main 4B setting, while larger-scale behavior is diagnostic of scale-dependent probe regimes.

τ	4B acc.	8B acc.	32B acc.
0.90	$16/24$ (66.7%)	$11/24$ (45.8%)	$9/24$ (37.5%)
1.20	$16/24$ (66.7%)	$16/24$ (66.7%)	$9/24$ (37.5%)
1.28	$\mathbf{17/24}$ (**70.8**%)	$\mathbf{18/24}$ (**75.0**%)	$10/24$ (41.7%)
1.36	$15/24$ (62.5%)	$15/24$ (62.5%)	$13/24$ (54.2%)
1.45	$16/24$ (66.7%)	$13/24$ (54.2%)	$14/24$ (58.3%)
1.55	$14/24$ (58.3%)	$13/24$ (54.2%)	$\mathbf{17/24}$ (**70.8**%)
1.65	$14/24$ (58.3%)	$13/24$ (54.2%)	$15/24$ (62.5%)

The purpose of this subsection: This subsection tests whether the main operating point $\tau = 1.28$ is a fragile choice. The sweep is a ***post-hoc sensitivity analysis***. It evaluates how accuracy changes after the rule is fixed, and it is not used to fit $C(x)$, (α, β, γ), or any dataset-specific decision.

Sweeping τ on the standard $[0.90, 1.40]$ grid with step 0.01 gives 4B accuracies in $[0.58, 0.71]$ and 8B diagnostic accuracies in $[0.46, 0.75]$. Tab. 14 additionally reports values at $\tau \in \{1.45, 1.55, 1.65\}$, included to inform the 32B bounded-benefit operating point selected in App. F.2. At 4B, the near-optimum plateau with accuracy at least $16/24$ spans $\tau \in [1.12, 1.31]$, and the peak plateau with accuracy $17/24$ spans $\tau \in [1.26, 1.31]$. The main operating point $\tau = 1.28$ lies inside the 4B peak plateau, and the reported 4B result is therefore not a brittle threshold artifact. We do not claim that $\tau = 1.28$ transfers as a universal constant across scales. The 8B and 32B columns of Tab. 14 are diagnostic, used to characterize how the same transport-efficiency framework behaves under scale-dependent probe regimes. Per-scale operating points are reported separately in App. F.2. Tab. 14 reports representative values from the sweep.

The main observation is that the 4B rule is locally flat around the chosen operating point. A ± 0.03 perturbation around $\tau = 1.28$ leaves the 4B accuracy unchanged at $17/24$. Across $\tau \in [1.20, 1.30]$, the 4B rule remains at or above $16/24$. This local flatness is a property of the 4B operating regime. The analogous plateau at 8B and 32B is shifted, not centered at $\tau = 1.28$. This stability follows the decision geometry of Sec. 3.2: because $\mathrm{TE}(x) = \mathrm{ISR}(x) \cdot \mathrm{VCR}(x)$ varies smoothly with the probes, small movements of the threshold contour in Fig. 3 reassign only borderline benchmarks.

The 32B column is included as a stress test. The 4B operating point does not transfer directly to 32B. At $\tau = 1.28$, the diagnostic 32B accuracy is only $10/24$. This is consistent with the scale-dependent probe behavior in Tab. 17: the small 32B precision weight α reduces the ability of $C(x)$ to penalize precision-sensitive inputs. As a result, text-friendly inputs can remain above the 4B threshold. The 32B result therefore motivates the bounded-benefit diagnostic extension in App. F.2, rather than contradicting the 4B rule.

F.2 OT Decision Accuracy across Scales

Table 15: OT decision-rule accuracy across model scales, with score and input-token impact relative to an always-LLM baseline. The 4B row uses the main operating point with $\tau = 1.28$. The 8B row applies the same threshold as a cross-scale diagnostic. The 32B row uses the diagnostic bounded-benefit extension with $\tau_{32B} = 1.55$. Score columns are macro-averages of per-dataset scores under each routing strategy. Token columns are macro-averages of mean input tokens per sample.

			Score		Input tokens		
Scale	τ	OT acc.	LLM → Routed	Δ score	LLM → Routed	Δ tok	Δ tok (%)
4B	1.28	$\mathbf{17/24}$ ($\mathbf{70.8}\%$)	45.9 → 47.4	+1.5 (+3.3%)	2092 → 1876	−216	−10.3%
8B	1.28	$\mathbf{18/24}$ ($\mathbf{75.0}\%$)	49.5 → 50.8	+1.3 (+2.6%)	2092 → 1861	−231	−11.0%
32B	1.55	$\mathbf{17/24}$ ($\mathbf{70.8}\%$)	51.1 → 52.5	+1.4 (+2.7%)	2092 → 1938	−154	− 7.4%

The purpose of this subsection: This subsection reports how the decision framework behaves as model scale changes. The 4B row corresponds to the main operating point reported in the paper. The 8B and 32B rows are diagnostic cross-scale operating points used to study how the same transport-efficiency framework behaves under scale-dependent probe regimes.

Scale-aware diagnostic extension at 32B. The main 4B operating point transfers well to the 8B diagnostic setting, but the same threshold overcommits to the visual path at 32B. The reason is not a change in the framework itself, but a change in the probe-calibrated cost parameters. At 32B, $\alpha = 0.053$ and $\gamma = 0.241$ (Tab. 17), so the cost term has little precision signal and the structured-format bonus becomes large. This pushes $\mathrm{TE}(x)$ upward on some text-friendly inputs. To diagnose this failure mode, we use a bounded compression-benefit variant at 32B. The standard rule is

$$\mathrm{TE}(x) = \big(1 + \gamma - C(x)\big) \cdot \mathrm{VCR}(x).$$

For the 32B diagnostic extension, we replace $\mathrm{VCR}(x)$ by

$$\mathrm{VCR}_{\mathrm{eff}}(x) = 1 + \min\big(\mathrm{VCR}(x) - 1,\ 0.30\big),$$

and compute

$$\mathrm{TE}_{32B}(x) = \big(1 + \gamma - C(x)\big) \cdot \mathrm{VCR}_{\mathrm{eff}}(x).$$

This caps extreme compression benefits when the cost term has lost precision signal. The decision rule remains a single threshold test, $\mathrm{TE}_{32B}(x) \geq \tau_{32B}$. We report $\tau_{32B} = 1.55$ as the operating point selected from the diagnostic sweep in Tab. 14. This extension is not part of main 4B claim. It is used only to analyze how framework behaves when the probe parameters become scale-degenerate.

Tab. 15 reports the per-scale diagnostic operating point of the framework and its score-token tradeoff. At 4B, the main operating point delivers a positive macro-average score gain of $+1.5$ points over the always-LLM baseline, while reducing macro-average input tokens by 10.3%. At 8B, the diagnostic rule reaches $18/24$ under the same $\tau = 1.28$ operating point, with a $+1.3$ point score gain and 11.0% fewer input tokens. At 32B, the diagnostic bounded-benefit rule recovers $17/24$ accuracy and yields a $+1.4$ point score gain at 7.4% fewer input tokens. The smaller token reduction at 32B reflects the conservative effect of bounding the compression benefit, which routes fewer datasets to the visual path than at 4B and 8B.

The 4B accuracy of $17/24$ matches the best supervised baseline reported in Tab. 3. At 8B, the diagnostic rule reaches $18/24$ under the same $\tau = 1.28$ operating point. The 32B result should be read as a scale-aware diagnostic rather than a replacement for the main downstream-label-free result. It shows that when probe parameters become degenerate, the same transport-efficiency framework identifies the source of failure and suggests a conservative correction.

Why does the token reduction vary non-monotonically with scale? The macro-average input-token reduction is -10.3% at 4B, -11.0% at 8B, and -7.4% at 32B. This non-monotone shape reflects which datasets the rule sends to the visual arm at each scale, not a change in per-sample visual-token efficiency. At 4B, the rule routes 9 of 24 datasets to the visual arm, including long-context QA and summarization datasets where each routed dataset saves several hundred tokens. At 8B, the diagnostic rule routes the same 9 datasets plus Yahoo, and several long-context visual

encodings become slightly more compact, so the macro reduction grows to -11.0%. At 32B, the bounded-benefit rule is more conservative and routes only the four lowest-cost multi-hop QA datasets. These routed datasets save many tokens individually, but the savings are averaged with 20 unrouted datasets, producing the smaller aggregate reduction of -7.4%. This trend is consistent with the framework: as the LLM strengthens, fewer inputs justify the visual path, but the inputs that remain selected are those where absolute token savings are largest.

F.3 External Path-Identification Diagnostic on VTCBench

Table 16: VTCBench subtask-level path identification under the 4B-calibrated $\tau = 1.28$. LLM and VLM columns report macro-average scores under the native `contains_all` metric. "Better path" is the higher-scoring of the two. The rule's majority routing matches the better path on the retrieval and reasoning subtasks. On the memory subtask, the high mean VCR pushes $\mathrm{TE}(x)$ above τ for nearly all samples, and the rule's majority routing disagrees with the better path. The non-tied accuracy column is reported on the subset of samples where the LLM and VLM scores strictly differ.

Subtask	n	mean VCR	LLM	VLM	rule $\rightarrow$ vis (%)	better path	subtask match	non-tied acc
Retrieval	800	1.44	99.4	71.8	39.6	text	✓	39.6% ($n_{\mathrm{nt}} = 280$)
Reasoning	800	1.51	29.9	6.6	38.6	text	✓	69.7% ($n_{\mathrm{nt}} = 234$)
Memory	600	4.87	30.7	15.3	96.3	text	×	5.8% ($n_{\mathrm{nt}} = 104$)

The purpose of this subsection: This subsection tests where the calibrated 4B routing rule remains directionally correct and where it fails on an external benchmark that did not participate in any parameter fit. The goal is not to claim a routed-score improvement on VTCBench, but to diagnose whether the rule's ***subtask-level decision***, defined as the path receiving the majority of routed samples, agrees with the empirically stronger path on each subtask.

Setup. We evaluate Qwen3-4B on VTCBench [52] with the same 4B parameters $(\alpha, \beta, \gamma) = (0.213, 0.627, 0.069)$ and the same operating point $\tau = 1.28$ used in the main paper. No VTCBench label, score, or subtask outcome is used to fit (α, β, γ), $C(x)$, τ, or any routing decision. $\mathrm{VCR}(x)$ is computed per sample as the ratio of LLM-side text-token count to VLM-side rendered-image-token count, and $\mathrm{TE}(x) = (1 + \gamma - C(x)) \cdot \mathrm{VCR}(x)$ follows Sec. 3.2. For each subtask we report the fraction of samples routed to the visual path, the empirically better path under macro LLM/VLM scores, whether the rule's majority routing matches that path, and the rule's accuracy on non-tied samples. The non-tied subset contains only samples where the LLM and VLM scores strictly differ, which avoids inflating accuracy with ties at 0 or 1.

What this shows. On retrieval and reasoning, the same operating point used in the main paper routes the majority of samples to the empirically stronger path, without any re-tuning on VTCBench. This is a subtask-level directional check rather than a routed-score result, since the table evaluates majority path identification and non-tied routing accuracy, not the final routed score. The reasoning subtask also clears the non-tied accuracy threshold of 50%, reaching 69.7%. Retrieval does not, reaching only 39.6% on non-tied samples. This means the retrieval row should be interpreted as a correct majority-path identification result, not as evidence of reliable per-sample discrimination. The non-tied column is included precisely to make this distinction explicit.

Memory exposes a boundary of the multiplicative TE form. On memory, the rule's majority routing disagrees with the empirically stronger path. The cause is structural rather than a result of missing threshold tuning. The mean visual-compression ratio is 4.87, with some samples reaching $\mathrm{VCR} > 40$, so the multiplicative form $\mathrm{TE}(x) = (1 + \gamma - C(x)) \cdot \mathrm{VCR}(x)$ can become large even when the text path is already near its score ceiling. A diagnostic sweep suggests that correcting memory would require $\tau \geq 3.0$, but such a threshold would also move retrieval and reasoning toward near always-text decisions. This sweep is diagnostic only and is not used to select any reported operating point. We therefore report memory as an external limitation of the current multiplicative TE form on near-ceiling, high-VCR text benchmarks. This limitation is consistent with the paper's broader claim: VTC is useful only when compression benefit outweighs task-relevant transport loss, and high compression alone is not sufficient.

G Limitations of Foveation at Scale

The purpose of this section: This section examines when transport-informed foveation helps and when it is limited. The goal is not to present foveation as a uniformly beneficial add-on, but to test whether its gains and limitations are consistent with the same scale-dependent cost parameters that drive the global routing rule.

In Sec. 3.3, foveation is defined as a localized inverse of the push-forward map, driven by the patch-level cost map

$$C_q = \alpha W_q + \beta L_q(1 - \mathrm{TRR}_q).$$

Because this local map uses the same scalars (α, β) as the global cost $C(x)$, its effectiveness should vary with the scale-specific probe regime. We therefore examine the calibrated probe parameters first, and then test whether the empirical FOV gains follow the same scale-dependent pattern.

G.1 Scale-Dependent Probe Regimes

Table 17: Per-scale calibrated parameters of $C(x) = \alpha W(x) + \beta L(x)(1 - \mathrm{TRR}(x))$ and the structured-format bonus γ. At 8B, the coverage weight β weakens, so C_q is dominated by the precision term. At 32B, the precision weight α weakens, so C_q is dominated by the coverage term.

Scale	α (intra)	β (inter)	γ (struct.)	Cost map regime
4B	0.213	0.627	0.069	both terms active
8B	0.455	**0.061**	0.000	precision-dominated: $C_q \approx \alpha W_q$
32B	**0.053**	0.233	0.241	coverage-dominated: $C_q \approx \beta L_q(1 - \mathrm{TRR}_q)$

The purpose of this subsection: This subsection documents the scale-specific probe parameters (α, β, γ) and identifies which transport-cost dimension remains reliable at each backbone scale. These parameters determine both the global cost $C(x)$ and the patch-level foveation map C_q.

The downstream-label-free probes calibrate (α, β, γ) separately at each scale, with the values reported in Tab. 17. Two asymmetric regimes emerge beyond 4B. At 4B, both α and β are non-trivial, so the cost map carries information from both within-patch precision variance and cross-patch coverage dispersion. Foveation can therefore identify either kind of high-cost region. At 8B, β becomes small, so the cost map is effectively precision-dominated and has little signal for identifying dispersed evidence patches. At 32B, the situation is reversed: α becomes small while β remains usable, so the cost map is coverage-dominated and is better suited to recovering evidence regions than fine-grained lexical distinctions. This scale dependence is not a separate property of foveation. It follows from the same probe-calibrated parameters that drive the global transport-efficiency rule.

G.2 Empirical FOV across Scales

Table 18: Per-dataset foveation gain at three scales. Each scale shows two columns: the baseline VLM score and the FOV minus VLM gap in absolute points. Positive Δ_{Fov} means foveation improves the visual path on that dataset. The All row reports paired averages over all 24 datasets at each scale after applying the trigger-zero convention in Sec. G.5.

	4B		8B		32B	
Dataset	VLM	Δ_{Fov}	VLM	Δ_{Fov}	VLM	Δ_{Fov}
Multi-choice RC						
RACE-H	25.1	$+0.81$	23.9	$+0.90$	26.7	-0.99
QuALITY	60.5	$+0.00$	67.2	$+0.96$	75.3	-0.34
LogiQA	38.9	$+0.46$	41.3	$+0.00$	42.9	$+0.00$
DREAM	71.1	$+0.05$	81.9	-0.34	85.4	$+0.44$
Binary						
BoolQ	78.3	$+0.00$	84.2	-0.40	87.2	$+0.40$
IMDB	82.4	$+0.00$	86.7	$+0.00$	85.7	$+0.00$
Multi-class						
DBpedia	93.8	$+0.00$	96.5	$+0.00$	97.3	$+0.00$
Yahoo	64.1	$+0.13$	66.7	-0.08	67.7	-0.03
Yelp	45.8	$+0.00$	53.2	$+0.00$	55.3	$+0.00$
NLI/verif.						
ContractNLI	41.9	$+0.68$	41.1	-0.77	41.3	-1.54
SciFact	71.4	$+0.84$	70.4	-0.05	74.9	$+0.00$
Short-answer QA						
HotpotQA	35.4	$+1.03$	37.7	-1.59	29.3	$+4.27$
MuSiQue	19.0	$+1.13$	22.8	-0.99	18.9	$+2.53$
2WikiMHQA	38.2	$+0.30$	44.0	-3.25	35.7	$+3.84$
CoQA	39.2	$+0.05$	48.6	$+0.11$	25.3	$+0.68$
DROP	22.0	$+0.00$	26.2	$+0.13$	21.1	$+0.23$
MultiFieldQA	38.8	$+0.41$	44.1	$+0.36$	41.7	$+1.24$
QASPER	30.1	$+2.93$	45.0	-0.61	31.7	$+0.70$
ReCoRD	26.5	$+0.52$	29.0	-0.53	46.8	$+0.04$
Summarization						
CNN/DM	24.1	$+0.00$	26.8	$+0.00$	28.2	$+0.00$
XSum	16.8	$+0.00$	17.4	$+0.00$	17.7	$+0.00$
BillSum	40.9	$+0.00$	41.2	$+0.00$	42.3	$+0.00$
GovReport	32.5	$+0.00$	37.1	$+0.00$	41.8	$+0.00$
QMSum	28.0	$+0.86$	29.0	-0.08	30.1	$+0.38$
All (paired n)	44.4	$\mathbf{+0.42}$	48.4	$\mathbf{-0.26}$	47.9	$\mathbf{+0.49}$

The purpose of this subsection: This subsection tests whether localized inverse transport improves visual path only when patch-level cost map contains the right signal for the task family. The foveation choices are made by the label-free cost map described in Sec. 3.3. VLM and FOV scores reported here are used only for post-hoc evaluation of whether those choices recover useful information.

Tab. 18 reports, for each of the 24 benchmarks, the VLM score without foveation and the FOV minus VLM gap at 4B, 8B, and 32B. A positive Δ_{Fov} means that foveation improves the visual path, while a negative value means that the selected crops add token cost or noise without recovering enough useful information. The aggregate paired gains are $\Delta_{\mathrm{Fov}} = +0.42$ at 4B, -0.26 at 8B, and $+0.49$ at 32B. These values are paired over all 24 datasets at each scale and apply the trigger-zero reporting convention in Sec. G.5. The non-monotonic trajectory is consistent with the scale-dependent probe regimes in Tab. 17. Foveation is beneficial at 4B, where both precision and coverage terms are active. It becomes negative at 8B, where the coverage weight β is weak and the cost map has little signal for selecting dispersed evidence regions. It becomes positive again at 32B, where the coverage signal becomes usable again and the largest gains concentrate on datasets whose evidence can be localized to a small number of useful regions.

The per-family pattern is more informative than the aggregate. *Short-answer QA* is the family with the clearest scale-dependent foveation behavior: its average foveation gain is positive at 4B, negative at 8B, and strongly positive at 32B. This family contains multi-hop and span-style QA tasks

where recoverable evidence often appears in localized regions of the rendered passage. When the coverage term is weak at 8B, foveation can select regions that are locally dense but not necessarily evidence-bearing. When the coverage signal becomes usable again at 32B, the same mechanism can identify evidence patches more effectively. In contrast, *Multi-choice RC* and *NLI/verif.* are more sensitive to precision. They become less favorable to foveation at 32B, where α is small and the cost map carries little precision signal. Summarization shows only small foveation gains despite being coverage-heavy, because the relevant information is often diffuse across long documents rather than concentrated in a few crop-worthy regions.

G.3 Discussion: Non-uniform Gains as Self-Consistency

The purpose of this discussion: This discussion interprets the non-uniform foveation gains as a scale-aware limitation of the method rather than as an unrelated failure. It connects the local behavior of C_q back to the global transport-cost structure $C(x)$.

The fact that foveation does not produce a uniform gain at every scale is a limitation of a scale-blind deployment, but it is also consistent with the proposed framework. The local refinement in Sec. 3.3 is, by construction, the patch-level instantiation of the same cost decomposition that drives the global decision in Sec. 3.2. When a probe-calibrated parameter becomes weak at a particular scale, the cost map loses sensitivity to the corresponding failure mode, and foveation can select regions that do not recover enough information to offset their token cost. The pattern in Tab. 18 therefore reads as follows. When the local cost map retains the signal needed by a task family, foveation contributes. When the relevant signal is weak, foveation contributes little or can hurt. This supports the claim that $C(x)$ and C_q are two linked instantiations of the same transport-cost structure rather than independent heuristics that only agree at 4B.

A practical consequence is that a scale-blind foveation deployment is not the right interface beyond the 4B setting. A better deployment is to gate foveation by the same scale-aware probes that gate routing. Foveation should be enabled only when the relevant cost dimension remains reliable at the target scale. We leave this scale-conditioned foveation gate to future work. The parameter table in Tab. 17 provides the information needed to implement it.

G.4 Sample-Level Visualization

The purpose of this subsection: The aggregate gains in Tab. 18 are summary statistics. This subsection complements them with qualitative sample-level evidence that the cost map localizes recoverable evidence and that the resulting crops can correct VLM-baseline errors. The examples are selected from datasets where foveation produces a sizeable positive macro gain, so they should be read as illustrative cases rather than as an unbiased estimate of average performance.

For visualization, we show the rendered page, a normalized patch-cost overlay, and the crop region selected by foveation. The displayed heatmap emphasizes the coverage component $\beta L_q(1 - \mathrm{TRR}_q)$ because it is directly interpretable from the rendered text. The deployed foveation rule, however, uses the full patch-level cost map C_q defined in Sec. 3.3. Thus, the figures should be interpreted as visual explanations of where the selected evidence lies, not as a separate decision rule.

Interpreting the hot patches. The cost-map overlay marks where question keywords concentrate after redundancy is discounted (BM25-style query-patch locality L_q damped by patch redundancy TRR_q), not the literal pixel location of the gold answer. The two regions coincide on extractive QA, where the gold answer string is co-located with question keywords on the visible page (the MultiFieldQA panels and the extractive QASPER s1 panel), but can diverge in two structural cases. First, for inference-based questions whose gold answer paraphrases or summarizes the supporting text (QASPER s2 and s3, both yes/no), hot patches mark the topic-relevant evidence the model needs to reason from, not the answer string itself. Second, for multi-hop questions whose disambiguating evidence sits on a later rendered page than the first page shown here (HotpotQA s2 and s3), hot patches highlight one branch of the multi-hop chain that lies on the visible page, while the gold-answer string may not appear on this page at all. In all cases, both the cost map and the selected crop are computed in the rendering coordinate system at inference time and are never selected with reference to the gold answer.

Visualized samples. We visualize three QASPER samples, three HotpotQA samples, and three MultiFieldQA samples. The QASPER and HotpotQA examples show cases where the baseline VLM scores at or near 0 and foveation lifts the prediction to 1. The MultiFieldQA examples show partial F1 improvements on long-form QA. Each sample yields one figure with three stacked panels: rendered page, patch-cost overlay, and selected crop box. The corresponding question, gold answer, and predicted strings are consolidated in Tab. 19, Tab. 20, and Tab. 21.

Table 19: QASPER foveation examples corresponding to Figs. 4–6. The QASPER evaluation pipeline records F1 only, so predicted text is omitted. Long strings are truncated.

Fig.	Question	Gold	F1 change
4	Which other tasks are evaluated?	product category classification and review headline generation	$0.26 \rightarrow 1.00$
5	Did they use crowdsourcing for the annotations?	No	$0.00 \rightarrow 1.00$
6	Are the recommendations specific to a region?	No	$0.00 \rightarrow 1.00$

Table 20: HotpotQA foveation examples corresponding to Figs. 7–9. The baseline VLM gives incorrect short answers, while VLM+Fov recovers the gold entity. Long strings are truncated.

Fig.	Question	Gold	VLM baseline	VLM+Fov
7	What pop band features an Argentine actress that appeared in the television series "Chiquititas"?	Erreway	F1 = 0.00: "The pop band is 'Tinys Angels'."	F1 = 1.00: "Erreway"
8	Which species is native to Asia, Corokia or Rhodotypos?	Rhodotypos	F1 = 0.00: "Corokia"	F1 = 1.00: "Rhodotypos"
9	Which was formed first, Noori or Test Icicles?	Noori	F1 = 0.00: "Test Icicles"	F1 = 1.00: "Noori"

Table 21: MultiFieldQA foveation examples corresponding to Figs. 10–12. VLM+Fov improves answer specificity by re-examining high-cost evidence regions. Long strings are truncated.

Fig.	Question	Gold	VLM baseline	VLM+Fov
10	Which air unit did Goodwin command during the initial landings of Marines on Saipan?	VC-10 Squadron	F1 = 0.22: "The air unit Goodwin commanded during the initial landings ... was VC-10 Squadron."	F1 = 1.00: "VC-10 Squadron"
11	What was the conclusion of the study?	The conclusion was that fruit consumption may provide a protective effect for mercury exposure ...	F1 = 0.26: "The document contains multiple studies on mercury exposure from fish consumption ..."	F1 = 0.58: "The conclusion of the study was that fruit consumption may provide a protective effect ..."
12	According to the text, what is Toby Schindelbeck's observation about the police?	Toby Schindelbeck's observation is that the police say they are not paid enough to enforce the laws ...	F1 = 0.48: "Toby Schindelbeck observes that the police are not paid enough to enforce the laws ..."	F1 = 0.52: "Toby Schindelbeck observes that the police are not paid enough to enforce the laws ..."



Figure 4: QASPER example 1. The three panels show the rendered page, the normalized patch-cost overlay, and the selected crop box used by foveation. Hot patches mark question-keyword concentration after redundancy is discounted, not literal answer locations.



Figure 5: QASPER example 2. The three panels show the rendered page, the normalized patch-cost overlay, and the selected crop box used by foveation. Hot patches mark question-keyword concentration after redundancy is discounted, not literal answer locations.



Figure 6: QASPER example 3. The three panels show the rendered page, the normalized patch-cost overlay, and the selected crop box used by foveation. Hot patches mark question-keyword concentration after redundancy is discounted, not literal answer locations.



Figure 7: HotpotQA example 1. The three panels show the rendered page, the normalized patch-cost overlay, and the selected crop box used by foveation. Hot patches mark question-keyword concentration after redundancy is discounted, not literal answer locations.



Figure 8: HotpotQA example 2. The three panels show the rendered page, the normalized patch-cost overlay, and the selected crop box used by foveation. Hot patches mark question-keyword concentration after redundancy is discounted, not literal answer locations.

HotpotQA



Figure 9: HotpotQA example 3. The three panels show the rendered page, the normalized patch-cost overlay, and the selected crop box used by foveation. Hot patches mark question-keyword concentration after redundancy is discounted, not literal answer locations.



Figure 10: MultiFieldQA example 1. The three panels show the rendered page, the normalized patch-cost overlay, and the selected crop box used by foveation. Hot patches mark question-keyword concentration after redundancy is discounted, not literal answer locations.



Figure 11: MultiFieldQA example 2. The three panels show the rendered page, the normalized patch-cost overlay, and the selected crop box used by foveation. Hot patches mark question-keyword concentration after redundancy is discounted, not literal answer locations.



Figure 12: MultiFieldQA example 3. The three panels show the rendered page, the normalized patch-cost overlay, and the selected crop box used by foveation. Hot patches mark question-keyword concentration after redundancy is discounted, not literal answer locations.

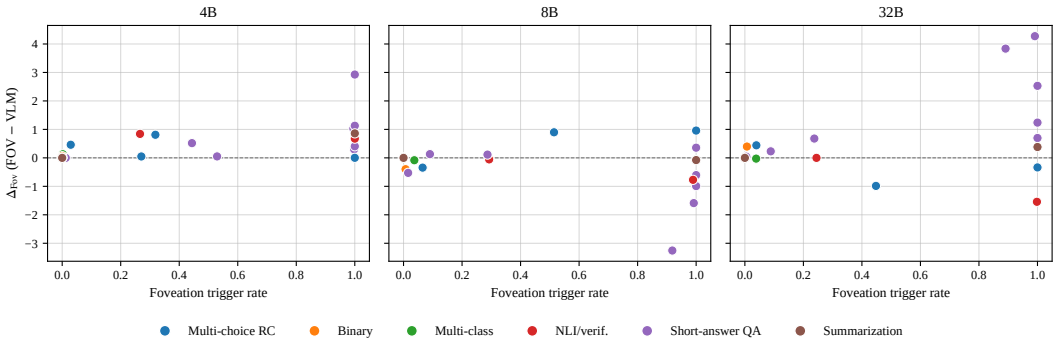


Figure 13: Per-dataset relationship between the foveation trigger rate and the foveation gain Δ_{Fov} at each backbone scale. Each point is one benchmark, colored by task family. Datasets with near-zero trigger rate cluster around $\Delta_{\mathrm{Fov}} = 0$ at every scale. Datasets with high trigger rate carry most of the scale-dependent positive and negative Δ_{Fov} changes, including the largest 32B short-answer-QA gains and the largest 8B short-answer-QA losses.

G.5 Foveation Trigger Statistics

The purpose of this subsection: The foveation gain Δ_{Fov} in Tab. 18 is a per-dataset macro statistic, but the underlying decision is made per sample. This subsection documents the per-sample trigger that gates whether an input is actually foveated. It also explains why datasets with near-zero trigger rate have $\Delta_{\mathrm{Fov}} \approx 0$ by construction, while high-trigger datasets carry most of the positive and negative scale-dependent changes.

Trigger definition. We say the foveation trigger fires on a sample x when the transport-efficiency inequality of Eq. 16 is satisfied. Equivalently, the relative cost recovered by the selected patches must exceed the relative token overhead:

$$\frac{\sum_k \Delta C_k}{\mathrm{ISR}(x)} > \frac{n_c}{n_v}, \qquad \mathrm{ISR}(x) = 1 + \gamma - C(x),$$

where the sum runs over the patches added by the greedy region-selection procedure of Sec. 3.3, and n_c, n_v are the additional and base visual-token counts. In practice, the trigger is evaluated only when the cost map $\{C_q\}$ contains a localized hot region. We use the criterion $\max_q C_q/\overline{C_q} > 2.5$. If the cost map is approximately uniform, no region can recover enough cost to justify extra visual tokens, so the sample is forwarded through the base VLM path. All quantities in the trigger are computed from probe-calibrated, downstream-label-free features.

Convention for trigger-zero benchmarks. If the trigger never fires on a benchmark, every sample is forwarded along the unfoveated VLM path by construction. In that case, the FOV pipeline and the VLM pipeline are identical for reporting purposes, and $\Delta_{\mathrm{Fov}} = 0$. We therefore report $\Delta_{\mathrm{Fov}} = +0.00$ on every scale-benchmark cell whose trigger rate is 0.0% in Tab. 22. This avoids interpreting small numerical differences from independent evaluation runs as foveation effects. The same convention is applied to Tab. 18, Fig. 13, and the aggregate statistics in this section.

Tab. 22 reports the per-dataset trigger rate at all three scales, alongside the per-dataset Δ_{Fov} that it produces. The trigger rate varies strongly across datasets. It is 0% on Binary tasks, most Multi-class tasks, and most short-summary tasks, but approaches 100% on QASPER, MuSiQue, MultiFieldQA, ContractNLI, QMSum, QuALITY, HotpotQA, and 2WikiMultiHopQA. The bimodal distribution is meaningful. When the cost map has no hot region, the trigger suppresses foveation. When the cost map identifies one or two coverage-cost peaks, which is typical of long-context QA and meeting summarization, nearly every sample triggers.

Foveation abstains on long-document summarization. A specific consequence of the trigger design is that foveation does *not* attempt to repair long-document summarization. On BillSum, XSum, GovReport, and CNN/DM the trigger rate is exactly 0% at all three scales, so the FOV pipeline reproduces the VLM pipeline by construction and $\Delta_{\mathrm{Fov}} = +0.00$. This behavior is consistent with the theory rather than an oversight. Summarization is the canonical *inter*-dominated regime: the

Table 22: Per-dataset foveation trigger rate at three scales together with the resulting per-dataset Δ_{Fov} from Tab. 18. The trigger rate is the fraction of samples for which the foveation trigger of Eq. 16 fires. The macro-average row averages over all 24 datasets at each scale.

	Trigger rate			Δ_{Fov}		
Dataset	4B	8B	32B	4B	8B	32B
Multi-choice RC						
RACE-H	31.9%	51.4%	44.8%	+0.81	+0.90	−0.99
QuALITY	100.0%	100.0%	100.0%	+0.00	+0.96	−0.34
LogiQA	2.9%	0.0%	0.0%	+0.46	+0.00	+0.00
DREAM	27.1%	6.5%	3.9%	+0.05	−0.34	+0.44
Binary						
BoolQ	0.0%	0.7%	0.7%	+0.00	−0.40	+0.40
IMDB	0.0%	0.0%	0.0%	+0.00	+0.00	+0.00
Multi-class						
Yahoo	0.2%	3.7%	3.9%	+0.13	−0.08	−0.03
DBpedia	0.0%	0.0%	0.0%	+0.00	+0.00	+0.00
Yelp	0.0%	0.0%	0.0%	+0.00	+0.00	+0.00
NLI/verif.						
SciFact	26.6%	29.3%	24.5%	+0.84	−0.05	+0.00
ContractNLI	100.0%	98.9%	99.8%	+0.68	−0.77	−1.54
Short-answer QA						
2WikiMHQA	99.8%	91.9%	89.1%	+0.30	−3.25	+3.84
HotpotQA	99.5%	99.2%	99.1%	+1.03	−1.59	+4.27
MuSiQue	100.0%	100.0%	100.0%	+1.13	−0.99	+2.53
DROP	1.2%	9.0%	8.8%	+0.00	+0.13	+0.23
QASPER	100.0%	100.0%	100.0%	+2.93	−0.61	+0.70
CoQA	52.9%	28.7%	23.7%	+0.05	+0.11	+0.68
MultiFieldQA	100.0%	100.0%	100.0%	+0.41	+0.36	+1.24
ReCoRD	44.4%	1.6%	0.5%	+0.52	−0.53	+0.04
Summarization						
BillSum	0.0%	0.0%	0.0%	+0.00	+0.00	+0.00
XSum	0.0%	0.0%	0.0%	+0.00	+0.00	+0.00
QMSum	100.0%	100.0%	100.0%	+0.86	−0.08	+0.38
GovReport	0.0%	0.0%	0.0%	+0.00	+0.00	+0.00
CNN/DM	0.0%	0.0%	0.0%	+0.00	+0.00	+0.00
Macro avg	41.1%	38.4%	37.5%	+0.42	−0.26	+0.49

relevant evidence is distributed across many rendered regions, so no single patch carries enough recoverable cost to satisfy the trigger inequality of Eq. 16. Adding a small number of higher-resolution patches cannot repair fragmentation that is spread across the entire document. We therefore interpret the zero-trigger summarization cells as a deliberate abstention, not as a degenerate failure: the rule chooses not to spend extra visual tokens when the cost map provides no localizable target. The lone summarization exception, QMSum, has a 100% trigger rate because its meeting-style transcripts concentrate the answer-relevant span around a small number of dialogue turns.

The scatter in Fig. 13 shows that Δ_{Fov} is bounded by the trigger rate. A dataset with trigger rate close to zero cannot have a large $|\Delta_{\mathrm{Fov}}|$ because the FOV pipeline degenerates to the VLM pipeline on most of its samples. This is visible in the scatter cloud near the origin at every scale. The non-trivial Δ_{Fov} values, both the positive ones at 4B and 32B and the negative ones at 8B, all originate from datasets in the high-trigger-rate region. The trigger therefore acts as a downstream-label-free filter that concentrates foveation on the inputs whose cost map signals it has something to recover. The empirical Δ_{Fov} behavior of Tab. 18 is consistent with this structure: positive at 4B where the cost map has both terms, negative at 8B where the coverage signal is weak in the relevant dataset population, and positive again at 32B where the coverage signal becomes usable again.

H Rendering Protocol and Font Size Sensitivity

Table 23: Font size sensitivity of the base visual path at three scales. Scores are macro-averages over the 24-benchmark suite without foveation. The default setting is 12 pt.

	All score			
Font size	4B	8B	32B	Tokens
font10	42.4	46.4	45.8	1,185
font12 (Default)	44.4	48.4	47.9	1,486
font14	45.3	49.7	49.8	1,716

The purpose of this section: This section documents the deterministic rendering protocol used to construct visual-path inputs and tests whether the visual-path behavior follows the expected font size tradeoff. The goal is not to tune rendering per dataset, but to verify that a single global rendering knob changes score and token count in a predictable way.

Rendering setup. For all visual-path runs, we render the input text as a single-column black-on-white RGB document image. The default setting uses Roboto-Regular at 12 pt, line spacing 1.0, no extra inter-line gap, and antialiased glyph rasterization. The page width and height are sized adaptively to fit the rendered text, then snapped to the encoder patch grid and capped at 928 pixels per dimension. Longer inputs are paginated, with interior pages reaching the full 928×928 pixel bound. The rendered image is passed to the Qwen3-VL visual encoder, which uses 16×16 raw patches followed by a 2×2 spatial merge inside the encoder.

Among these rendering choices, font size directly controls the average number of characters that fall into one visual patch. It therefore changes the strength of within-patch aggregation T_{intra} in the push-forward map. Within our transport view, smaller fonts pack more characters into each patch and should increase lexical smoothing, while larger fonts reduce smoothing at the cost of more visual tokens. We test this prediction by sweeping font size in $\{10, 12, 14\}$ while keeping the font family, layout, color scheme, page sizing rule, patch size, and spatial merge factor fixed. The default 12 pt setting is used globally in all primary experiments.

The sweep shows a consistent score-token tradeoff at all three scales. Shrinking the font from 12 pt to 10 pt reduces the visual token count but lowers the macro score. Increasing the font to 14 pt raises the macro score but increases token cost. The score change from 10 pt to 14 pt is $+2.9$ points at 4B, $+3.3$ points at 8B, and $+4.0$ points at 32B. The direction is the same at every scale.

This behavior supports the transport interpretation. Font size is not a hidden tuning parameter that changes unpredictably across tasks. It acts as a global rendering knob controlling the precision-token tradeoff induced by within-patch aggregation. The fixed 12 pt setting used in the main experiments provides a balanced operating point between smaller, more compressed renderings and larger, less compressed renderings. Because the same global rendering rule is applied to all datasets and scales, the reported routing and foveation results are not produced by dataset-specific rendering choices.

I Runtime and Memory Measurements

Table 24: Runtime and peak-memory measurements on six representative benchmarks at the 4B scale. Latency is reported as seconds per sample. Peak memory is measured as GPU memory in GB. The routed strategy uses the label-free routing decision and reports the measured latency and memory of the selected path. The macro-average row is the unweighted average over the six datasets.

		Latency (sec/sample)				Peak memory (GB)			
Dataset	Length regime	LLM	VLM	FOV	Routed	LLM	VLM	FOV	Routed
RACE-H	short	0.16	0.12	0.13	0.12	8.7	9.3	9.3	9.3
BoolQ	short	0.08	0.13	0.13	0.08	8.2	8.8	8.8	8.2
DBPedia	short	0.05	0.11	0.11	0.05	8.1	8.7	8.7	8.1
SciFact	medium	0.18	0.17	0.18	0.17	8.8	9.3	9.4	9.3
HotpotQA	medium	1.16	0.41	0.42	0.41	9.6	10.2	10.2	10.2
GovReport	long	9.59	6.38	6.34	6.38	12.9	22.4	22.4	22.4
Macro avg	—	1.87	1.22	1.22	1.20	9.4	11.5	11.5	11.3

The purpose of this section: This section complements the token-count analysis in the main paper with wall-clock runtime and GPU-memory measurements. The goal is not to claim that the visual path is always faster, but to test whether the token reductions reported in the main results can translate into practical latency reductions on representative short-, medium-, and long-context benchmarks.

Protocol. To obtain implementation-level runtime measurements, we instrument the evaluation pipeline by wrapping `transformers.GenerationMixin.generate` to record per-call wall-clock time using `time.perf_counter` and peak GPU memory using `torch.cuda.max_memory_allocated`, without changing the generation outputs. We measure on six benchmarks selected to span the input-length spectrum of the 24-benchmark suite: RACE-H (multi-choice RC, short), BoolQ (binary, short), DBPedia (multi-class, short), SciFact (claim verification, medium), HotpotQA (multi-hop QA, medium), and GovReport (long-document summarization, long). All runs use Qwen3-4B/Qwen3-VL-4B-Instruct with 12 pt rendering, on a single NVIDIA L40S GPU, in the same offline-inference configuration as the main experiments. For the routed strategy, we compute the label-free routing decision first and then assign the measured latency of the selected path. This avoids re-running identical generations and does not use task labels or model scores. The routing computation itself is a lightweight CPU-side calculation and is not included in the table.

What the latency table shows. Three patterns are worth highlighting. First, on ***short-context*** tasks (RACE-H, BoolQ, DBPedia), all strategies sit within 0.05–0.16 sec/sample. At this length range, neither the text nor the visual path uniformly dominates wall-clock latency. The routed strategy selects text for BoolQ and DBPedia and visual for RACE-H under the label-free TE rule, yielding latency close to the faster measured path on these short-context rows. Second, on ***longer-context*** tasks (HotpotQA, GovReport), the visual path is meaningfully faster than the LLM path: 0.41 vs. 1.16 sec/sample on HotpotQA ($2.8\times$ faster) and 6.38 vs. 9.59 sec/sample on GovReport ($1.5\times$ faster). Long inputs make the LLM decoder pay for many text tokens, while the rendered visual representation reduces the effective sequence length. Third, the ***macro-average*** routed latency is 1.20 sec/sample, a 36% reduction relative to the LLM baseline (1.87). It is also slightly lower than the always-visual baseline (1.22), although this difference is small after rounding. Thus, the routed strategy preserves much of the runtime benefit of the visual path on inputs where it is selected, while reverting to the cheaper text path when rendering plus visual encoding would add overhead.

Foveation cost. On the six datasets in Tab. 24, FOV differs from VLM by at most 0.04 sec/sample. This small runtime difference reflects that foveation uses a bounded crop budget and is activated only for selected regions. The runtime of FOV is therefore close to the runtime of the base visual path on this representative subset, and it does not change the macro-level latency picture.

Memory. Peak GPU memory is dominated by the longest input rather than by the routing rule. On GovReport, the VLM path retains rendered pages and visual tokens in memory, reaching a peak

of 22.4 GB, while the LLM path peaks at 12.9 GB. On all other datasets, the two paths are within 0.5–1.0 GB of each other. Because foveation uses a bounded crop budget and does not retain extra crops beyond the base visual representation, FOV peak memory is within 0.1 GB of VLM in every case. None of the strategies exceeded the available memory on the L40S GPU in this measurement.

Interpretation. These measurements support the efficiency claim made in the main text. Token reduction is not merely an accounting statistic: on longer-context inputs, it translates into lower wall-clock latency. At the same time, the measurements also show why routing is necessary. The visual path is not uniformly faster on short inputs, and it can require more memory on long rendered documents. The label-free decision rule therefore plays a practical role beyond accuracy, selecting the visual path where its token savings are likely to matter while retaining the text path where rendering overhead would dominate.

J Cross-Architecture Validation: InternVL3.5

The purpose of this section: This section tests whether the task-dependent pattern of visual-versus-text behavior described by the framework is specific to the Qwen3-VL family used in the headline experiments, or whether it transfers to a different visual encoder. We evaluate ***InternVL3.5*** [44], a vision-language model that pairs an InternViT visual encoder with the same Qwen3 LLM backbones used in our main experiments. The 4B and 8B InternVL3.5 models use Qwen3-4B and Qwen3-8B respectively, and the 38B model uses Qwen3-32B (verified from the released checkpoint configuration). This pairing is convenient because it keeps the LLM backbone matched at every scale and provides a diagnostic view of how changing the visual encoding stack affects the visual path. The text path through the LLM backbone is identical between Qwen3-VL-xB and the InternVL3.5 variant that shares the same Qwen3 LLM, so visual-path differences primarily reflect the visual encoder.

Protocol. We evaluate InternVL3.5 using the same evaluation harness as the main experiments, adapting only the model-loading and inference wrappers required by InternVL. Every benchmark uses identical prompt templates, scoring code, and rendering parameters (Roboto-Regular, 12 pt, adaptive page sizing capped at 928 px) as the Qwen3-VL runs. The only deviation is that InternVL's native multi-image API (`chat(num_patches_list=...)`) is used to handle paginated inputs, which performs the model-specific dynamic tiling expected by InternVL. We evaluate the full 24-benchmark suite at three scales: Qwen3-VL-4B / InternVL3.5-4B (Qwen3-4B LLM), Qwen3-VL-8B / InternVL3.5-8B (Qwen3-8B LLM), and Qwen3-VL-32B / InternVL3.5-38B (Qwen3-32B LLM). Foveation is not run on InternVL because the current foveation implementation assumes Qwen3-VL's patch grid and ViT feature layout. We therefore restrict the cross-architecture comparison to baseline visual paths. At each scale we use the scale-specific visual-friendly partition: S_2 at scale σ is the set of datasets where the visual arm (max of VLM and FOV) wins LLM under the bolding rule of Tab. 13 at that scale. This gives $|S_2| = 10$ at 4B, $|S_2| = 8$ at 8B, and $|S_2| = 11$ at 32B. The per-scale composition shifts as the LLM backbone scales, as described below.

Table 25: Macro-average scores at the Qwen3-4B / InternVL3.5-4B scale. "LLM" is text-only Qwen3-4B. "Qwen-VL" is Qwen3-VL-4B, "+Fov" is Qwen-VL with foveation, and "InternVL" is InternVL3.5-4B baseline. Δ_{Int} is InternVL minus Qwen-VL. The S_2/S_3 split is the 4B visual-friendly partition (10 datasets visual-wins, 14 datasets text-wins, See Tab. 13).

Group	LLM	Qwen-VL	+Fov	InternVL	Δ_{Int}
All ($n=24$)	45.9	44.4	44.8	46.9	+2.5
S_2 visual-friendly at 4B ($n=10$)	34.7	44.0	44.7	43.8	−0.2
S_3 text-friendly at 4B ($n=14$)	54.0	44.6	44.8	49.1	+4.5
Multi-choice RC ($n=4$)	57.9	48.9	49.2	52.9	+4.0
Binary ($n=2$)	90.0	80.4	80.4	91.9	+11.5
Multi-class ($n=3$)	65.5	67.9	68.0	68.8	+0.9
NLI/verif. ($n=2$)	51.4	56.7	57.4	57.1	+0.4
Short-answer QA ($n=8$)	28.2	31.1	31.9	33.9	+2.8
Summarization ($n=5$)	33.1	28.5	28.6	27.5	−0.9

Macros are within ± 2.5 points on the full suite. Tab. 25, Tab. 26, and Tab. 27 report macros at the 4B, 8B, and 32B scales; Tab. 28 reports the per-dataset Δ_{Int} values that back the per-dataset claims below. On the full 24-benchmark macro, InternVL leads Qwen-VL by $+2.5$ at 4B, $+0.9$ at 8B, and $+0.9$ at 32B. The per-dataset win-loss counts of InternVL versus Qwen-VL are 11 / 11 / 2 ties at 4B, 12 / 12 / 0 at 8B, and 11 / 9 / 4 ties at 32B (Tab. 28, $|\Delta_{\text{Int}}| \leq 0.5$ tie threshold), so the two architectures are evenly matched at the dataset level at every scale even when the macro is close. The framework's main qualitative finding, that VTC behavior is sharply task-dependent, replicates under InternViT at all three scales: on both architectures some tasks favor the visual path, others favor the text path, and compression ratio alone does not predict which case applies.

The visual-friendly partition itself is scale-dependent. $|S_2|$ shifts non-monotonically across scales: 10 at 4B, 8 at 8B, and 11 at 32B. The transitions are interpretable from the cost decomposition. From 4B to 8B, `race` and `drop` flip to text-wins because the 8B LLM closes the precision-sensitive gap on these tasks. From 8B to 32B, `race`, `multifieldqa`, and `qmsum` enter S_2. `race` re-enters

Table 26: Macro-average scores at the Qwen3-8B / InternVL3.5-8B scale. The S_2/S_3 split is the 8B visual-friendly partition: S_2 at 8B contains 8 datasets where the visual path wins under the bolding rule of Tab. 13 at 8B. This is the 4B set minus `drop` and `race`, since both flip to S_3 at 8B as the 8B LLM closes the gap. All scores are in percent.

Group	LLM	Qwen-VL	+Fov	InternVL	Δ_{Int}
All ($n=24$)	49.5	48.4	48.2	49.3	+0.9
S_2 visual-friendly at 8B ($n=8$)	44.5	53.0	52.2	47.6	−5.4
S_3 text-friendly at 8B ($n=16$)	52.0	46.1	46.1	50.2	+4.1
Multi-choice RC ($n=4$)	56.2	53.6	54.0	54.2	+0.7
Binary ($n=2$)	90.8	85.4	85.2	92.5	+7.0
Multi-class ($n=3$)	68.4	72.1	72.1	70.2	−1.9
NLI/verif. ($n=2$)	57.9	55.8	55.3	62.9	+7.2
Short-answer QA ($n=8$)	37.2	37.2	36.4	38.3	+1.1
Summarization ($n=5$)	32.5	30.3	30.3	27.8	−2.5

Table 27: Macro-average scores at the Qwen3-32B / InternVL3.5-38B scale. Both models share the same Qwen3-32B LLM backbone, so visual-path differences primarily reflect the InternViT-vs-Qwen-VL encoder swap. The S_2/S_3 split is the 32B visual-friendly partition: S_2 at 32B contains 11 datasets where the visual arm (max of VLM and FOV) wins LLM under the bolding rule of Tab. 13 at 32B. Compared to 8B, `race`, `multifieldqa`, and `qmsum` enter S_2, while `drop` remains outside S_2 as the 32B LLM closes the precision gap. All scores are in percent.

Group	LLM	Qwen-VL	+Fov	InternVL	Δ_{Int}
All ($n=24$)	51.1	47.9	48.4	48.8	+0.9
S_2 visual-friendly at 32B ($n=11$)	40.8	45.1	46.2	42.2	−2.9
S_3 text-friendly at 32B ($n=13$)	59.7	50.3	50.3	54.3	+4.0
Multi-choice RC ($n=4$)	65.8	57.6	57.3	58.5	+0.9
Binary ($n=2$)	91.8	86.5	86.7	93.6	+7.1
Multi-class ($n=3$)	73.1	73.4	73.4	72.9	−0.5
NLI/verif. ($n=2$)	62.5	58.1	57.3	56.2	−1.9
Short-answer QA ($n=8$)	33.2	31.3	33.0	34.0	+2.7
Summarization ($n=5$)	33.7	32.0	32.1	29.2	−2.8

because the baseline visual path slightly exceeds the LLM at 32B, while `multifieldqa` and `qmsum` enter because the 32B foveation regime restores a usable coverage signal (App. G). `drop` stays out as the 32B LLM remains stronger on this precision-sensitive task. Eight datasets are visual-friendly at every scale and form the stable core of S_2: `2wikimhqa`, `hotpotqa`, `scifact`, `musique`, `yahoo`, `dbpedia`, `billsum`, and `qasper`. We therefore split S_2/S_3 per scale rather than imposing a single partition uniformly.

InternVL gives back where Qwen-VL leads, and recovers where Qwen-VL trails. The most informative comparison is the per-scale S_2/S_3 rows. The same direction holds at every scale: InternVL underperforms Qwen-VL on S_2 where Qwen-VL has a strong visual lead, and outperforms Qwen-VL on S_3 where Qwen-VL is below the LLM. At 4B the pattern is mild (S_2: -0.2, S_3: $+4.5$), at 8B it is most pronounced (S_2: -5.4, S_3: $+4.1$), and at 32B it remains visible (S_2: -2.9, S_3: $+4.0$). Two dataset families illustrate this contrast. Long-document QA, which tends to be visual-friendly under Qwen-VL (`qasper`, `multifieldqa`, `hotpotqa`), shows large InternVL deficits at 4B (-14.8, -12.0, -10.5; Tab. 28). Precision-sensitive cloze and reasoning tasks where Qwen-VL trails the LLM (`drop`, `record`, `dream`) show large InternVL gains. The two ViT families therefore have different but partially complementary strengths at every scale we tested. This suggests that Qwen-VL's native visual encoder is better aligned with cross-page coverage on long documents, while InternViT appears stronger on short precision-sensitive inputs.

Table 28: Per-dataset cross-architecture gain $\Delta_{\mathrm{Int}} = \mathrm{InternVL} - \mathrm{Qwen\text{-}VL}$ at three scales. Positive values mean InternVL outperforms Qwen-VL on that dataset. The 32B column compares Qwen3-VL-32B against InternVL3.5-38B (both share the Qwen3-32B LLM). Tab. 25, Tab. 26, and Tab. 27 report the macro and family-level aggregates from the same data. The win/loss/tie counts cited in this section use a tie threshold of $|\Delta_{\mathrm{Int}}| \leq 0.5$. Per-dataset Δ_{Int} values are rounded to one decimal from full-precision differences. Their 24-row average can differ from the macro Δ_{Int} in Tabs. 25–27 by up to ± 0.1 when the full-precision macro lies near a 1dp display boundary. This affects the 32B row of the macro tables: the 24-row average of the per-dataset values shown here is $+0.83$, which rounds to $+0.8$, while subtracting the 1dp macros in Tab. 27 gives $+0.9$.

Dataset	4B Δ_{Int}	8B Δ_{Int}	32B Δ_{Int}
Multi-choice RC			
RACE-H	$+1.1$	$+2.2$	-0.1
QuALITY	-9.6	-12.4	-16.3
LogiQA	$+9.7$	$+7.7$	$+13.8$
DREAM	$+14.8$	$+5.2$	$+6.2$
Binary			
BoolQ	$+9.7$	$+5.1$	$+3.2$
IMDB	$+13.4$	$+8.9$	$+11.1$
Multi-class			
DBpedia	-1.2	-4.0	-3.1
Yahoo	-7.5	-8.8	-6.2
Yelp	$+11.5$	$+7.1$	$+7.7$
NLI/verif.			
ContractNLI	-1.1	$+9.0$	-2.7
SciFact	$+2.0$	$+5.4$	-1.1
Short-answer QA			
HotpotQA	-10.5	-6.3	-0.5
MuSiQue	-4.3	-2.3	$+3.5$
2WikiMHQA	-5.5	-5.1	$+2.6$
CoQA	$+15.1$	$+5.1$	$+6.3$
DROP	$+38.6$	$+36.1$	$+9.8$
MultiFieldQA	-12.0	-15.7	-16.9
QASPER	-14.8	-21.8	-3.7
ReCoRD	$+15.5$	$+18.7$	$+20.3$
Summarization			
CNN/DM	$+7.6$	$+3.8$	$+4.0$
XSum	$+0.4$	-0.6	-0.5
BillSum	-0.2	-0.7	-0.5
GovReport	-5.9	-10.6	-10.6
QMSum	-6.6	-4.4	-6.4

Cross-architecture intervention changes a different axis from foveation. At 4B, switching the visual encoder from Qwen3-VL's native ViT to InternViT yields $+2.5$ points on the full macro, compared with the $+0.4$ foveation gain reported in App. G. At 8B, foveation is slightly negative (-0.3, consistent with the scale-dependent probe regimes of App. G), while the InternVL swap still yields $+0.9$. However, the per-dataset profiles of the two interventions are very different. Foveation produces small, sample-conservative changes bounded by its trigger gate, with the largest 4B gains on evidence-localized QA tasks such as `qasper` ($+2.9$), `hotpotqa` ($+1.0$), and `musique` ($+1.1$). The InternVL swap is a high-variance intervention (Tab. 28): $+38.6$ on `drop` and $+15.5$ on `record` at 4B, but -14.8 on `qasper` and -12.0 on `multifieldqa` at the same scale. This asymmetry is consistent with the two operations being conceptually different. Foveation refines the within-patch resolution of a fixed ViT according to a probe-derived cost map. Switching architectures replaces the entire push-forward map with a differently trained one. The two are not substitutes.

Per-family pattern is consistent with the cost decomposition. The family rows of Tab. 25, Tab. 26, and Tab. 27 show structured differences that hold across scales. Binary classification favors InternVL substantially at every scale ($+11.5$ at 4B, $+7.0$ at 8B, $+7.1$ at 32B), and multi-choice RC and short-answer QA favor InternVL by smaller margins. Summarization favors Qwen-VL at every scale (-0.9, -2.5, -2.8), in line with summarization being inter-cost-dominated and harder to repair by either intervention. NLI/verif. is less stable across scales, moving from an InternVL advantage at 4B and 8B to a Qwen-VL advantage at 32B. We do not claim a quantitative explanation of these per-family differences. We report them so that readers using a different VLM family can anticipate where the visual path's behavior may shift.

What this section does and does not claim. This section reports a sample-paired cross-architecture replication. The result we want to convey is that the framework's main object, a task-dependent visual-versus-text gap whose ordering is largely set by intra-patch precision and inter-patch coverage, is reproduced under a different ViT family at three backbone scales. We do not claim that $C(x)$ predicts the exact magnitude of the InternVL gap on each dataset, since the magnitudes are architecture-specific. We also do not claim that InternVL3.5 should replace Qwen3-VL as the headline backbone. The two are within ± 2.5 macro points, with a balanced win-loss split, and the choice between them is a deployment decision rather than a research claim. What the cross-architecture data adds is direct evidence against the worry that VTC task-dependent behavior is an artifact of one specific ViT. The same pattern appears under InternViT, with predictable family-level differences.