FROM TARGET-WEIGHTED COVERAGE TO CONDITIONAL INFORMATION FOR VISUAL TOKEN PRUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) process long sequences of visual tokens, resulting in substantial inference cost. Existing visual token pruning methods typically rely on instruction relevance, visual redundancy, or their combination, but can retain tokens whose evidence relevant to the instruction is already represented by the selected set. We propose *TACI* (Target-Aware Coverage and Conditional Information), a training-free framework that selects visual evidence according to the additional information it provides about the instruction beyond the tokens already retained. Once the instruction representations have incorporated visual context, *TACI* evaluates each candidate using conditional mutual information. To avoid carrying the full visual sequence until this point, it first constructs a compact candidate pool using target-weighted visual coverage. Both stages use Gaussian process (GP) models over token representations, providing a unified probabilistic framework for the selection process. Experiments on seven benchmarks with LLaVA and Qwen backbones demonstrate the effectiveness of *TACI* across different token budgets.

1 INTRODUCTION

Recent vision-language models (VLMs) can perform a broad range of visual understanding and reasoning tasks (Liu et al., 2023; 2024a; Bai et al., 2025). However, processing visual inputs requires the language model to handle hundreds or even thousands of additional tokens, resulting in substantial inference latency and memory overhead. This problem becomes more severe for high-resolution images, multi-image inputs, and videos, where visual tokens can dominate the input sequence (Bai et al., 2025; Khaki et al., 2025; Liu et al., 2025). The resulting computational burden makes VLMs difficult to deploy in applications requiring low latency, long visual contexts, or limited computational resources. Visual token pruning addresses this challenge by removing redundant or uninformative visual tokens before or during language-model inference (Zhang et al., 2025c; Yang et al., 2025; Dhouib et al., 2025).

Existing visual token pruning methods select tokens based on instruction relevance, visual redundancy, or their combination (Zhang et al., 2025c; Yang et al., 2025; Zhang et al., 2025b). Attention or text-guided approaches prioritize visual tokens that are strongly associated with the instruction (Zhang et al., 2025c; Chen et al., 2024a), while redundancy- and diversity-based methods seek to remove visually similar tokens and preserve a more representative subset (Yang et al., 2025; Wen et al., 2025). More recent approaches combine these signals, for example by optimizing instruction-conditioned diversity (Zhang et al., 2025b) or applying visual redundancy reduction and text-guided pruning at different stages of the model (Singh et al., 2026).

Despite these advances, existing approaches do not explicitly account for redundancy in the information relevant to the instruction: A candidate can remain relevant to the instruction even when the evidence it provides is already represented by the retained tokens. Similarly, visual diversity does not necessarily imply that a token contributes new evidence for the instruction. In such cases, a limited token budget may be spent on tokens that are individually relevant or visually distinct but provide little additional information beyond the selected set. This motivates evaluating each candidate not only by its relevance or visual diversity, but also by how much new information it contributes about the instruction given the tokens already retained, with this contribution reassessed as the selected set grows.

We propose *TACI* (Target-Aware Coverage and Conditional Information), a training-free framework for visual token pruning. *TACI* directly addresses this objective by selecting visual evidence according to the additional information it provides about the instruction beyond the tokens already retained.

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

Once the instruction representations have incorporated visual context, it evaluates each candidate using conditional mutual information. To avoid carrying the full visual sequence until this point, *TACI* first constructs a compact candidate pool using target-weighted visual coverage. Both stages are implemented using Gaussian process (GP) models over token representations, providing a unified probabilistic framework for the two-stage selection process.

Across seven vision-language benchmarks using LLaVA-1.5 and Qwen2.5-VL, *TACI* performed comparably to or better than state-of-the-art approaches across different token budgets and it was particularly effective under aggressive token reduction, demonstrating its potential for efficient VLM inference under limited computational resources.

2 Related work

Vision-language models (VLMs). Modern VLMs connect vision encoders to language models to support visual understanding and reasoning. LLaVA uses a learned projector and visual instruction tuning to align the two components (Liu et al., 2023; 2024a), while Qwen2.5-VL supports dynamic-resolution visual processing (Bai et al., 2025). Their visual inputs can produce long decoder sequences, motivating methods that reduce the number of visual tokens processed during inference.

Visual importance and instruction guidance. Visual token pruning methods preserve salient or representative image content, use instruction guidance, or combine the two. Some methods retain dominant tokens while removing or merging visually similar content (Yang et al., 2025; Zhang et al., 2025a; Dhouib et al., 2025), while others encourage diversity or coverage of the visual-token population (Alvar et al., 2025; Deng et al., 2025; Zou et al., 2025). Instruction-guided methods estimate token importance from visual-text associations such as decoder attention or textual relevance (Chen et al., 2024a; Xing et al., 2025; Zhang et al., 2025c).

Combining relevance and visual preservation. Some recent approaches combine instruction guidance with visual diversity, redundancy reduction, or coverage. DART uses both visual and textual pivots to identify duplicated content (Wen et al., 2025). CDPruner combines instruction-based relevance with diversity-aware subset selection (Zhang et al., 2025b), while MoB balances prompt alignment with preservation of the visual population (Li et al., 2025). These methods jointly consider task relevance and visual representativeness. *TACI* further assesses how much information a candidate contributes given the evidence already selected.

Multi-stage visual token pruning. Pruning location affects both the information available for selection and the amount of computation that can be saved. Pruning before the decoder reduces computation throughout subsequent layers, but selection must rely on relatively coarse visual and instruction representations. Pruning inside the decoder can exploit representations that have already incorporated visual-text interactions. DUET-VLM performs pruning at both locations by reducing visual tokens before and inside the decoder (Singh et al., 2026). *TACI* also adopts a two-stage selection design, using target-weighted coverage before the decoder and conditional information selection at an intermediate layer.

3 Visual token pruning with conditional information

TACI performs visual token selection in two stages. The first forms a compact candidate set using target-weighted coverage before decoder processing, while the second uses conditional mutual information (CMI) after the instruction representations incorporate visual context.

3.1 Problem formulation

A vision encoder E extracts features from an image I, and a projector g maps them into the language-model embedding space, producing N_0 visual tokens:

$$g(E(I)) = (\boldsymbol{v}_1^0, \ldots, \boldsymbol{v}_{N_0}^0), \quad \mathbb{V}^0 = \{\boldsymbol{v}_1^0, \ldots, \boldsymbol{v}_{N_0}^0\}.$$

At decoder layer l, the visual and instruction token sets are

$$\mathbb{V}^l = \{\boldsymbol{v}_1^l, \ldots, \boldsymbol{v}_{N_l}^l\}, \quad \mathbb{T}^l = \{\boldsymbol{t}_1^l, \ldots, \boldsymbol{t}_M^l\},$$

where $\boldsymbol{v}_i^l, \boldsymbol{t}_j^l \in \mathbb{R}^D$ and D is the hidden dimension of the decoder. Given a budget $K^l < N_l$, a pruning operator $\mathcal{P}$ selects a subset of the current visual tokens of size K^l:

$$\mathbb{S}^l = \mathcal{P}(\mathbb{V}^l, \mathbb{T}^l; K^l).$$

After selection, decoder processing continues with the tokens in $\mathbb{S}^l$ in their original sequence order, while all text tokens are preserved. We design $\mathcal{P}$ while keeping the vision encoder, projector, and language model fixed.

For a decoder with L blocks, the earlier coverage stage operates on $\mathbb{V}^0$ before the first decoder block and produces a candidate set of size K^0. Following the pruning schedule of DUET-VLM (Singh et al., 2026), we apply conditional selection at $l = \lfloor L/2 \rfloor$ and remove the remaining visual tokens at $m = \lfloor 3L/4 \rfloor$, without applying an additional selection criterion at m.

3.2 Conditional information selection

At decoder layer l, the candidate set $\mathbb{V}^l$ contains the visual tokens retained by the early reduction stage after passing through the lower decoder blocks. The instruction representations $\mathbb{T}^l$ have meanwhile incorporated visual context. We therefore use $\mathbb{T}^l$ as the target and select visual tokens according to the additional information they provide about the instruction given the tokens already selected. For inputs where the image precedes the question, instruction tokens can attend to preceding visual tokens, while visual tokens cannot attend to the later instruction. Therefore, $\mathbb{T}^l$ provides a target representation that incorporates visual context for conditional selection.

Shared token kernel. For any nonzero token representation $\boldsymbol{x}$, we define $\phi(\boldsymbol{x}) = \boldsymbol{x}/\|\boldsymbol{x}\|$ and use the normalized exponential kernel

$$\kappa(\boldsymbol{x}_a, \boldsymbol{x}_b) = \exp\big(\phi(\boldsymbol{x}_a)^\top \phi(\boldsymbol{x}_b) - 1\big). \tag{1}$$

Since $\phi(\boldsymbol{x})$ has unit norm, $\|\phi(\boldsymbol{x}_a) - \phi(\boldsymbol{x}_b)\|^2 = 2 - 2\phi(\boldsymbol{x}_a)^\top \phi(\boldsymbol{x}_b)$, so κ is equivalent to a Gaussian kernel with unit bandwidth. The same kernel is used in both selection stages. Only the token representations to which it is applied differ.

Joint covariance. To measure statistical dependence between visual and instruction tokens, we construct a covariance matrix over their joint token set $\mathbb{X}^l = \mathbb{V}^l \cup \mathbb{T}^l$. The covariance combines two sources of information: similarity between the decoder representations of the tokens and similarity derived from their attention queries and keys.

The first component is computed from the decoder representations:[1]

$$[\boldsymbol{B}^{\mathrm{hid}}]_{a,b} = \kappa(\boldsymbol{x}_a^l, \boldsymbol{x}_b^l). \tag{2}$$

The second uses the attention projections of the decoder. At attention head $h \in \{1, \ldots, H\}$, where H is the number of heads, each visual token $\boldsymbol{v}_i^l \in \mathbb{V}^l$ is projected to a key $\boldsymbol{k}_i^h$, while each instruction token $\boldsymbol{t}_j^l \in \mathbb{T}^l$ is projected to a query $\boldsymbol{q}_j^h$. This reflects the causal attention pattern, where instruction queries can attend to preceding visual keys. We use the query and key projections before applying rotary positional embeddings (RoPE), so that the resulting covariance reflects content similarity without the effect of positional rotation.

For each head, we associate every token in $\mathbb{X}^l$ with the corresponding query or key representation:

$$\boldsymbol{u}_a^h = \begin{cases} \boldsymbol{k}_i^h, & \text{if } \boldsymbol{x}_a^l = \boldsymbol{v}_i^l \in \mathbb{V}^l, \\ \boldsymbol{q}_j^h, & \text{if } \boldsymbol{x}_a^l = \boldsymbol{t}_j^l \in \mathbb{T}^l, \end{cases} \quad [\boldsymbol{B}^{\mathrm{QK}}]_{a,b} = \frac{1}{H} \sum_{h=1}^{H} \kappa(\boldsymbol{u}_a^h, \boldsymbol{u}_b^h). \tag{3}$$

This covariance matrix captures both the compatibility between visual keys and instruction queries and the similarity within the visual-key and instruction-query representations.

For $\alpha \in [0,1]$, we combine the two covariance matrices as

$$\boldsymbol{B} = (1-\alpha)\boldsymbol{B}^{\mathrm{hid}} + \alpha \boldsymbol{B}^{\mathrm{QK}}. \tag{4}$$

This convex combination is a symmetric positive semidefinite matrix (Section A).

[1] Throughout the rest of this paper, $[\boldsymbol{A}]_{i,j}$ represents a scalar entry of a matrix $\boldsymbol{A}$, $\boldsymbol{A}_{\mathbb{S},\mathbb{T}}$ represents the submatrix indexed by sets $\mathbb{S}$ and $\mathbb{T}$, and $\boldsymbol{A}_{:,\mathbb{T}}$ represents the columns indexed by $\mathbb{T}$.

Conditional information gain. We use $\boldsymbol{B}$ as the covariance matrix of a zero-mean Gaussian process (GP) over the joint token representations. Let $\mathbf{f} \sim \mathcal{N}(0, \boldsymbol{B})$ represent the latent variables, and let $\mathbf{Z} = \mathbf{f} + \boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \mathcal{N}(0, \varepsilon \boldsymbol{I})$, be the noisy variables used for conditioning. Their covariance matrix is $\boldsymbol{\Sigma} = \boldsymbol{B} + \varepsilon \boldsymbol{I}$. We use $\mathbf{Z}_{\mathbb{T}^l}$, the variables corresponding to the instruction tokens, as the target for conditional selection.

Given a selected subset $\mathbb{S}^l$ of visual tokens, we condition the GP on $\mathbf{Z}_{\mathbb{S}^l}$ and let $\boldsymbol{C}$ be the resulting posterior covariance matrix of the latent variables. The matrix $\boldsymbol{C}$ describes the uncertainty that remains after accounting for the selected tokens. For a candidate $\boldsymbol{v}_i^l \in \mathbb{V}^l \setminus \mathbb{S}^l$, we define $\boldsymbol{c}_i = \boldsymbol{C}_{\mathbb{T}^l, i}$ as the posterior covariance vector between the candidate and the instruction tokens under this conditioning.

We measure the additional information provided by the candidate using CMI:

$$\Delta_{\mathrm{CMI}}(\boldsymbol{v}_i^l \,|\, \mathbb{S}^l) = I_{\boldsymbol{\Sigma}}(\mathbf{Z}_i; \mathbf{Z}_{\mathbb{T}^l} \,|\, \mathbf{Z}_{\mathbb{S}^l}) = -\tfrac{1}{2}\log(1-\rho_i^2), \quad \rho_i^2 = \frac{\boldsymbol{c}_i^\top (\boldsymbol{C}_{\mathbb{T}^l, \mathbb{T}^l} + \varepsilon \boldsymbol{I})^{-1} \boldsymbol{c}_i}{[\boldsymbol{C}]_{i,i} + \varepsilon}. \tag{5}$$

Here, ρ_i^2 measures the association that remains between the candidate and the instruction tokens after conditioning on the visual tokens already selected. A larger value indicates that the candidate provides more information about the instruction beyond that represented by $\mathbb{S}^l$.

At each step, we select the candidate with the largest gain and update the posterior covariance matrix. The token representations and the prior covariance matrix remain fixed, while the posterior covariance matrix changes after each selection. The gain of each remaining candidate is recomputed after every selection.

Residual information. For the fixed Gaussian model, we define the residual information $\mathcal{R}(\mathbb{S}^l)$ as the information about the instruction that remains in the unselected visual tokens after conditioning on the selected subset:

$$\mathcal{R}(\mathbb{S}^l) = I_{\boldsymbol{\Sigma}}\big(\mathbf{Z}_{\mathbb{T}^l}; \mathbf{Z}_{\mathbb{V}^l \setminus \mathbb{S}^l} \,|\, \mathbf{Z}_{\mathbb{S}^l}\big) = I_{\boldsymbol{\Sigma}}(\mathbf{Z}_{\mathbb{T}^l}; \mathbf{Z}_{\mathbb{V}^l}) - I_{\boldsymbol{\Sigma}}(\mathbf{Z}_{\mathbb{T}^l}; \mathbf{Z}_{\mathbb{S}^l}). \tag{6}$$

For a candidate $\boldsymbol{v}_i^l \in \mathbb{V}^l \setminus \mathbb{S}^l$, the chain rule gives

$$\mathcal{R}(\mathbb{S}^l) - \mathcal{R}(\mathbb{S}^l \cup \{\boldsymbol{v}_i^l\}) = I_{\boldsymbol{\Sigma}}(\mathbf{Z}_i; \mathbf{Z}_{\mathbb{T}^l} \,|\, \mathbf{Z}_{\mathbb{S}^l}) = \Delta_{\mathrm{CMI}}(\boldsymbol{v}_i^l \,|\, \mathbb{S}^l). \tag{7}$$

Each CMI step therefore selects the candidate that produces the largest reduction in the residual information. Conditioning discounts information about the instruction that is already represented by the selected subset. A visually distinct candidate can therefore receive little gain if it provides limited new information about the instruction.

3.3 Early candidate reduction by target-weighted coverage

Conditional selection benefits from representations that have incorporated interaction between visual and text tokens, but retaining all visual tokens until decoder layer l would increase computation. *TACI* therefore forms a compact candidate pool before the decoder. At this stage, the projected visual and instruction tokens share the language model embedding space but have not yet interacted through the decoder. Their similarity provides a coarse measure of association that influences preservation, but the candidate selection itself is driven by visual coverage.

Target-weighted visual coverage. We use the shared kernel to construct the visual covariance matrix $\boldsymbol{K} \in \mathbb{R}^{N_0 \times N_0}$:

$$[\boldsymbol{K}]_{i,j} = \kappa(\boldsymbol{v}_i^0, \boldsymbol{v}_j^0). \tag{8}$$

For a selected subset $\mathbb{S}^0$, the posterior covariance matrix is

$$\boldsymbol{C}^{\mathbb{S}^0} = \boldsymbol{K} - \boldsymbol{K}_{:,\mathbb{S}^0} (\boldsymbol{K}_{\mathbb{S}^0,\mathbb{S}^0} + \varepsilon \boldsymbol{I})^{-1} (\boldsymbol{K}_{:,\mathbb{S}^0})^\top, \quad \boldsymbol{C}^{\emptyset} = \boldsymbol{K}. \tag{9}$$

The diagonal entry $[\boldsymbol{C}^{\mathbb{S}^0}]_{u,u}$ measures the uncertainty remaining for visual token u after conditioning on the selected subset.

To prioritize visual content associated with the instruction, we compute pairwise similarities between the visual and instruction tokens:

$$a_{ij} = \phi(\boldsymbol{v}_i^0)^\top \phi(\boldsymbol{t}_j^0).$$

Algorithm 1 *TACI*: Visual token pruning with conditional information. The complete algorithm is provided in Algorithm 2.

Input: visual tokens $\mathbb{V}^0$, instruction tokens $\mathbb{T}^0$
Parameters: budgets K^0, K^l, selection layer $l=\lfloor L/2 \rfloor$, final pruning layer $m=\lfloor 3L/4 \rfloor$
Target-weighted visual coverage
Construct visual covariance $\boldsymbol{K}$ and relevance weights $\{w_i\}_{i=1}^{N_0}$
$\mathbb{S}^0 \leftarrow \emptyset$
while $|\mathbb{S}^0| < K^0$ **do**
  Select $\boldsymbol{v}_{i^*}^0 = \underset{\boldsymbol{v}_i^0 \in \mathbb{V}^0 \setminus \mathbb{S}^0}{\operatorname{argmax}} \Delta_{\mathrm{cov}}(\boldsymbol{v}_i^0 \mid \mathbb{S}^0)$
  $\mathbb{S}^0 \leftarrow \mathbb{S}^0 \cup \{\boldsymbol{v}_{i^*}^0\}$
end while
Propagate the retained tokens to layer l to obtain $\mathbb{V}^l$ and $\mathbb{T}^l$
Conditional information selection
Construct the mixed covariance $\boldsymbol{B}$
$\mathbb{S}^l \leftarrow \emptyset$
while $|\mathbb{S}^l| < K^l$ **do**
  Select $\boldsymbol{v}_{i^*}^l = \underset{\boldsymbol{v}_i^l \in \mathbb{V}^l \setminus \mathbb{S}^l}{\operatorname{argmax}} \Delta_{\mathrm{CMI}}(\boldsymbol{v}_i^l \mid \mathbb{S}^l)$
  $\mathbb{S}^l \leftarrow \mathbb{S}^l \cup \{\boldsymbol{v}_{i^*}^l\}$
end while
Continue with the retained visual tokens until m, then remove all remaining visual tokens
Continue decoding with the text tokens

Then the relevance of visual token i is defined as the average of its top $q=\min(4,M)$ similarities with the instruction tokens:

$$r_i = \frac{1}{q} \sum_{j \in \mathrm{Top}_q(\{a_{ij}\}_{j=1}^{M})} a_{ij}, \tag{10}$$

where Top_q returns the indices of the q largest similarities. Collecting the relevance scores as $\boldsymbol{r}=[r_1,\ldots,r_{N_0}]^\top$, we define the normalized relevance weight

$$w_i = \frac{r_i - \min(\boldsymbol{r})}{\max(\boldsymbol{r}) - \min(\boldsymbol{r})}.$$

The weighted remaining uncertainty is

$$\mathcal{U}(\mathbb{S}^0) = \sum_{u=1}^{N_0} (w_u + \varepsilon_r)[\boldsymbol{C}^{\mathbb{S}^0}]_{u,u}. \tag{11}$$

Here, $\varepsilon_r > 0$ ensures a positive effective weight for all visual tokens. A smaller $\mathcal{U}(\mathbb{S}^0)$ indicates better visual coverage, with greater emphasis on content associated with the instruction.

Coverage-based selection. For a candidate $\boldsymbol{v}_i^0 \in \mathbb{V}^0 \setminus \mathbb{S}^0$, its coverage gain is the reduction in weighted uncertainty:

$$\Delta_{\mathrm{cov}}(\boldsymbol{v}_i^0 \mid \mathbb{S}^0) = \mathcal{U}(\mathbb{S}^0) - \mathcal{U}(\mathbb{S}^0 \cup \{\boldsymbol{v}_i^0\}) = \frac{\sum_{u=1}^{N_0} (w_u + \varepsilon_r) \big([\boldsymbol{C}^{\mathbb{S}^0}]_{u,i}\big)^2}{[\boldsymbol{C}^{\mathbb{S}^0}]_{i,i} + \varepsilon}. \tag{12}$$

At each step, we select the token $\boldsymbol{v}_{i^*}^0$ with the largest coverage gain, update the posterior covariance matrix, and set $\mathbb{S}^0 \leftarrow \mathbb{S}^0 \cup \{\boldsymbol{v}_{i^*}^0\}$. Selection continues until $|\mathbb{S}^0| = K^0$. The covariance update and derivation of the marginal gain are presented in Section A.2. Algorithm 1 summarizes the overall *TACI* procedure.

Computational complexity. The computational cost of *TACI* consists of the selection overhead and the decoder computation after pruning. Stage 1 selection costs $O(N_0^2 D + N_0 M D + K^0 N_0^2)$. Using exact rank-one Cholesky downdates, Stage 2 selection can be implemented with complexity

$$O\big((K^0+M)^2 D + (K^0+M) D^2 + M^3 + K^l \big[M^2 K^0 + (K^0+M)^2\big]\big). \tag{13}$$

Table 1: Performance on LLaVA-1.5-7B under different token-reduction budgets. The best and second-best results for each benchmark are highlighted in **bold** and underlined, respectively.

Method	GQA	MME	POPE	MMB	MMStar	VizWiz	SEED
Full visual tokens (100.0%)							
Original	61.89	1814.28	87.01	64.52	33.73	53.65	65.72
Average 192 tokens (↓ 66.67%)							
FastV (Chen et al., 2024a)	57.92	1785.81	81.34	63.57	32.20	**55.24**	63.14
PyramidDrop (Xing et al., 2025)	**61.88**	1804.39	86.73	64.69	**33.80**	53.74	**65.49**
HiMAP (Yin et al., 2025)	56.30	1726.27	81.12	61.08	31.93	53.85	59.27
SparseVLM (Zhang et al., 2025c)	60.26	1754.98	86.99	63.57	33.40	53.17	64.52
VisionZip (Yang et al., 2025)	59.06	1710.50	86.14	63.23	33.07	53.59	62.71
CDPruner (Zhang et al., 2025b)	60.52	1761.87	86.67	63.06	32.73	54.50	64.05
TransPrune (Li et al., 2026)	60.65	1768.59	86.67	**64.86**	32.67	52.19	64.80
DUET-VLM (Singh et al., 2026)	59.83	1740.68	**87.16**	64.00	32.87	51.83	64.76
TACI (Ours)	60.82	**1811.07**	87.03	63.32	32.73	53.75	65.02
Average 128 tokens (↓ 77.78%)							
FastV (Chen et al., 2024a)	55.64	1701.41	77.08	63.23	32.33	55.42	61.35
PyramidDrop (Xing et al., 2025)	60.45	1797.53	86.50	63.92	**34.40**	53.48	64.18
HiMAP (Yin et al., 2025)	54.42	1642.37	77.86	59.45	31.53	53.81	56.90
SparseVLM (Zhang et al., 2025c)	58.68	1724.37	86.22	63.49	33.00	54.08	63.34
VisionZip (Yang et al., 2025)	57.18	1693.81	85.00	62.29	32.67	54.33	61.32
CDPruner (Zhang et al., 2025b)	59.87	1733.72	**87.97**	62.03	32.40	**55.50**	63.20
TransPrune (Li et al., 2026)	59.95	1772.49	86.87	**64.43**	32.53	51.63	63.89
DUET-VLM (Singh et al., 2026)	58.68	1728.36	86.76	62.97	32.20	51.49	63.79
TACI (Ours)	**60.47**	**1834.87**	86.43	62.29	32.80	54.44	**64.31**
Average 64 tokens (↓ 88.89%)							
FastV (Chen et al., 2024a)	51.71	1495.55	69.88	59.54	31.47	55.24	55.92
PyramidDrop (Xing et al., 2025)	57.43	1647.45	81.67	62.37	32.87	52.44	58.69
HiMAP (Yin et al., 2025)	50.09	1390.37	68.87	54.90	30.67	53.95	51.78
SparseVLM (Zhang et al., 2025c)	55.75	1659.73	84.94	61.86	**33.73**	53.43	60.77
VisionZip (Yang et al., 2025)	54.90	1636.30	81.08	59.62	31.13	55.10	57.29
CDPruner (Zhang et al., 2025b)	58.93	1731.37	**87.60**	60.22	30.73	**57.04**	61.55
TransPrune (Li et al., 2026)	58.44	1746.27	86.30	**63.49**	32.67	51.81	62.40
DUET-VLM (Singh et al., 2026)	55.67	1706.02	84.58	62.29	32.20	51.29	60.88
TACI (Ours)	**59.18**	**1769.94**	84.91	60.57	31.80	55.68	**63.00**

Let M_{all} be the number of text tokens in the initial decoder input, and define $\mathcal{C}(n) = nD^2 + n^2D$, which is the decoder cost of one block for a sequence of length n. The decoder processes $K^0 + M_{\text{all}}$ tokens for the first l blocks, $K^l + M_{\text{all}}$ tokens for the next $m - l$ blocks, and only the text tokens for the remaining $L - m$ blocks. The resulting complexity is

$$O\big(l\mathcal{C}(K^0 + M_{\text{all}}) + (m-l)\mathcal{C}(K^l + M_{\text{all}}) + (L-m)\mathcal{C}(M_{\text{all}})\big). \tag{14}$$

Detailed selection and decoder computation costs are discussed in Section B.

4 Experiments

4.1 Experimental setting

We evaluate *TACI* using two VLM backbones, LLaVA-1.5-7B and Qwen2.5-VL-7B-Instruct. All model parameters are fixed. The experiments cover seven benchmarks spanning diverse aspects of visual understanding and reasoning. GQA (Hudson & Manning, 2019) focuses on compositional reasoning over object attributes and relations in real-world scenes, while VizWiz (Gurari et al., 2018) evaluates visual question answering under challenging real-world conditions, such as blur, poor lighting, and

Table 2: Performance on Qwen2.5-VL under different token-reduction budgets. Redundant 64-token results for PyramidDrop and TransPrune are not reported (see Section D for details).

Method	GQA	MME	POPE	MMB	MMStar	VizWiz	SEED
Full visual tokens (100.0%)							
Original	61.16	2353.27	87.72	84.71	64.93	71.81	76.81
Average 192 tokens							
FastV (Chen et al., 2024a)	56.46	2241.07	81.04	83.42	55.73	65.87	62.26
PyramidDrop (Xing et al., 2025)	56.89	2047.80	84.56	81.87	52.93	54.33	47.00
HiMAP (Yin et al., 2025)	55.86	2273.78	78.94	81.87	54.73	68.01	65.84
SparseVLM (Zhang et al., 2025c)	55.84	2282.13	80.91	81.96	56.53	64.76	65.42
VisionZip (Yang et al., 2025)	60.23	2310.73	82.94	83.59	61.13	68.74	74.05
CDPruner (Zhang et al., 2025b)	**61.03**	2268.01	83.94	83.33	58.53	**69.69**	75.63
TransPrune (Li et al., 2026)	58.03	1944.56	84.66	81.70	54.87	54.35	45.83
DUET-VLM (Singh et al., 2026)	52.43	**2327.58**	**87.80**	**85.48**	**62.27**	55.62	74.64
TACI (Ours)	60.39	2297.59	87.74	84.62	60.60	56.08	**75.85**
Average 128 tokens							
FastV (Chen et al., 2024a)	52.27	2023.74	76.93	77.66	48.73	57.57	52.61
PyramidDrop (Xing et al., 2025)	39.52	1191.48	60.79	55.58	40.07	52.78	46.41
HiMAP (Yin et al., 2025)	52.19	2147.19	75.67	80.15	51.80	63.14	59.16
SparseVLM (Zhang et al., 2025c)	51.25	2138.69	77.66	80.67	52.07	56.72	58.24
VisionZip (Yang et al., 2025)	58.70	2243.69	81.03	83.51	58.27	67.18	71.61
CDPruner (Zhang et al., 2025b)	**60.47**	2227.06	83.97	83.42	57.40	**67.84**	74.26
TransPrune (Li et al., 2026)	38.43	1065.96	50.03	40.12	36.20	52.63	44.40
DUET-VLM (Singh et al., 2026)	52.22	**2259.36**	87.12	84.02	59.27	55.61	72.96
TACI (Ours)	60.42	2256.59	**87.13**	**84.62**	**59.40**	56.12	**75.38**
Average 64 tokens							
FastV (Chen et al., 2024a)	39.37	998.80	52.61	42.70	37.27	52.63	41.40
PyramidDrop (Xing et al., 2025)				N/A			
HiMAP (Yin et al., 2025)	43.26	1395.18	63.09	65.38	42.40	52.78	41.80
SparseVLM (Zhang et al., 2025c)	42.61	1398.58	58.51	60.74	38.80	54.71	54.82
VisionZip (Yang et al., 2025)	54.74	2052.75	76.57	81.27	52.53	63.47	66.52
CDPruner (Zhang et al., 2025b)	**59.01**	2038.90	82.17	80.15	53.00	**64.30**	71.67
TransPrune (Li et al., 2026)				N/A			
DUET-VLM (Singh et al., 2026)	50.79	2074.82	85.78	81.36	55.00	55.72	69.79
TACI (Ours)	58.94	**2167.56**	**85.93**	**83.51**	**56.93**	55.57	**73.53**

framing issues. MME (Fu et al., 2025) covers 14 perception and cognition tasks, including object existence, counting, position, color, and commonsense reasoning. MMBench (Liu et al., 2024b) assesses a broad range of vision-language abilities through multiple-choice questions, while MMStar (Chen et al., 2024b) emphasizes visually indispensable questions to reduce language-only shortcuts. SEED-Bench (Li et al., 2024) evaluates multiple perception and reasoning dimensions, while POPE (Li et al., 2023) evaluates object hallucination through binary object-presence questions. We follow the evaluation settings commonly used in prior visual token pruning work (see Section D.2 for details).

We compare against eight training-free visual token reduction methods. FastV (Chen et al., 2024a) relies on instruction-guided visual importance, while VisionZip (Yang et al., 2025) focuses on reducing visual redundancy. We also compare against methods that combine instruction relevance with visual preservation, including PyramidDrop (Xing et al., 2025), HiMAP (Yin et al., 2025), SparseVLM (Zhang et al., 2025c), CDPruner (Zhang et al., 2025b), TransPrune (Li et al., 2026), and DUET-VLM (Singh et al., 2026), using different pruning stages or selection criteria. These baselines cover approaches based on relevance alone, visual redundancy alone, and explicit combinations of the two. In contrast, *TACI* evaluates each candidate by the additional instruction-related information it contributes given the visual evidence already retained, directly accounting for redundancy in instruction-relevant information. We use the same three token budgets throughout the main comparisons (see Section D.1 for details).

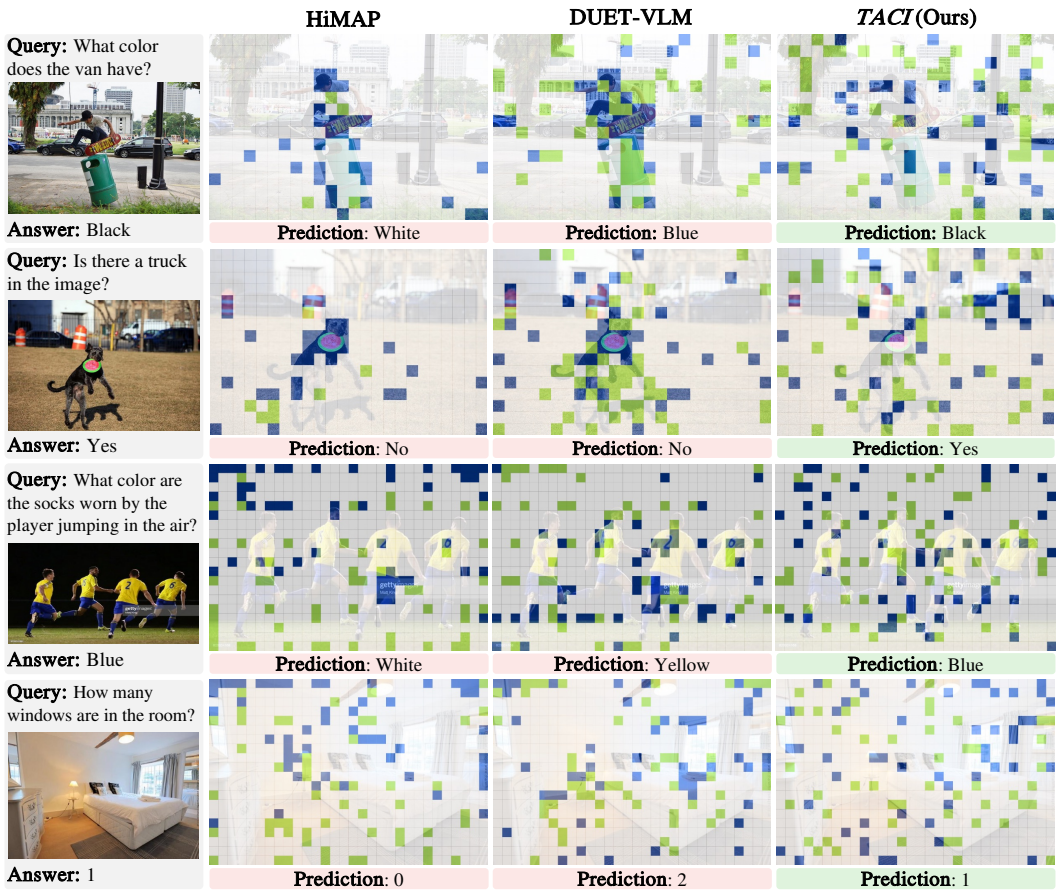

Figure 1: Examples of visual evidence retained by different token pruning methods. For *TACI*, tokens selected by target-weighted coverage are highlighted in green and blue. The tokens retained after conditional information selection are highlighted in blue. Top two rows: LLaVA-1.5 on GQA and POPE, respectively. Bottom two rows: Qwen2.5-VL on SEED-Bench.

4.2 Main Results

Table 1 presents the results on LLaVA-1.5-7B across the three token budgets. *TACI* maintains competitive performance across benchmarks. Its relative performance becomes stronger at lower token budgets, where the differences among pruning methods also become more pronounced. At the two higher budgets, the best results are distributed across several methods. At 128 tokens, *TACI* achieves the best results on GQA, MME, and SEED-Bench, and it retains the best results on the same three benchmarks at 64 tokens.

A similar trend is observed on Qwen2.5-VL (Table 2). At 192 tokens, *TACI* achieves the best result on SEED-Bench and the second-best results on GQA, POPE, and MMBench. Its relative performance becomes stronger as the visual-token budget decreases. At 128 tokens, *TACI* achieves the best results on POPE, MMBench, MMStar, and SEED-Bench and the second-best results on GQA and MME. At 64 tokens, it achieves the best results on MME, POPE, MMBench, MMStar, and SEED-Bench and the second-best result on GQA. VizWiz remains the main exception, where its performance is lower.

Figure 1 shows examples of the visual tokens retained by different pruning methods. In the first two rows, the questions require visual evidence about the van and the truck, respectively. HiMAP and DUET-VLM retain many tokens around the person in the first example and the person or dog in the second, while *TACI* allocates more of the retained tokens to visual evidence relevant to the question. In the first example, the relatively sparse evidence retained around the van may make its color more difficult to identify, with both methods predicting white or blue instead of black. In the second example, their limited coverage of the truck region may help explain why both methods predict that no truck is present, while *TACI* retains more evidence around the vehicle and gives the correct answer. In the third row, where the question asks about the socks worn by the player jumping in the air, *TACI* retains tokens across the relevant people while pre-

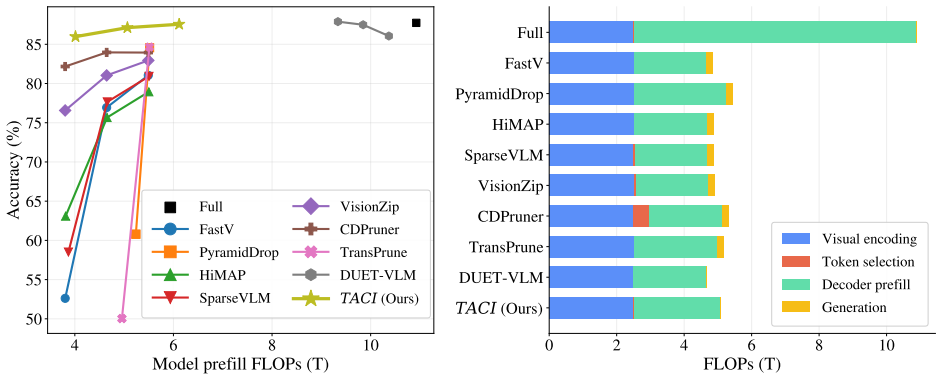

Figure 2: Performance-computation trade-off and FLOPs decomposition (POPE). (a) Relative performance and FLOPs across three token budgets. (b) FLOPs spent on visual encoding, selection, decoder computation, and generation. Both panels use a common device, input protocol, and cache policy.

serving evidence around the queried region. In the final row, *TACI* retains visual evidence around the window region and correctly predicts the number of windows, whereas both methods give incorrect answers.

Another visible pattern is that *TACI* tends to avoid excessive concentration of the tokens retained after conditional information selection within a single region. Tokens from such a region may each be individually relevant to the instruction, but once part of this evidence has been retained, additional nearby tokens can provide limited new information. This behavior is consistent with the conditional selection criterion, which discounts information already represented by the selected set and can therefore favor complementary visual evidence over repeated selection from the same region.

4.3 Inference Efficiency

We evaluate inference efficiency in terms of computational cost, including visual encoding, token selection, decoder computation, and generation. Figure 2 compares the performance computation trade-off across the three token budgets. All methods are evaluated under the same inference configuration.

TACI introduces additional computation for covariance construction and incremental selection. However, Stage 1 reduces the visual sequence before the decoder and also limits the candidate pool used by Stage 2. At the middle token budget, *TACI* reduces the model FLOPs by 3.13$\times$ compared with the full model.

5 Conclusion and Limitations

In this paper, we studied how to preserve visual information while reducing the number of visual tokens processed by vision-language models. We presented *TACI*, a training-free method that selects visual tokens according to the additional information they provide about the instruction beyond the evidence already retained. By conditioning on the selected tokens, *TACI* explicitly accounts for redundancy in instruction-relevant information. Target-weighted visual coverage provides a compact candidate pool before conditional selection, reducing the amount of visual information that must be carried through the decoder. Under the Gaussian process (GP) surrogate, each conditional mutual information selection corresponds to a reduction in residual target information. Across seven benchmarks and multiple token budgets, *TACI* maintains competitive performance.

Our current algorithm uses fixed locations for candidate generation and conditional selection. Future work could explore adapting these locations to individual inputs. The GP surrogate provides a systematic way to model dependence and redundancy among visual tokens and between visual and instruction tokens, but it does not directly model answer correctness. The incremental selection procedure chooses the candidate with the largest marginal gain at each step but does not optimize future selections jointly and therefore does not guarantee a globally optimal subset.

AI USE STATEMENT

In this work, we used generative AI tools for proofreading and revising the manuscript, verifying mathematical and algorithmic derivations, generating portions of the initial code, and identifying additional related work. We reviewed all AI-assisted content. In particular, LLM-generated initial code was verified, revised, and tested by the authors. We take responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 describes the proposed method, and the appendices provide the derivations and detailed experimental settings. The code will be released upon acceptance.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based visual token pruning for large multimodal models. In *CVPR*, 2025.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *ECCV*, 2024a.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, et al. Are we on the right way for evaluating large vision-language models? In *NeurIPS*, 2024b.

Jinhong Deng, Wen Li, Joey Tianyi Zhou, and Yang He. SCOPE: Saliency-coverage oriented token pruning for efficient multimodel LLMs. In *NeurIPS*, 2025.

Mohamed Dhouib, Davide Buscaldi, Sonia Vanier, and Aymen Shabou. PACT: Pruning and clustering-based token reduction for faster visual language models. In *CVPR*, 2025.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, et al. MME: A comprehensive evaluation benchmark for multimodal large language models. In *NeurIPS*, 2025.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. VizWiz grand challenge: Answering visual questions from blind people. In *CVPR*, 2018.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019.

Samir Khaki, Junxian Guo, Jiaming Tang, Shang Yang, Yukang Chen, Konstantinos N Plataniotis, Yao Lu, Song Han, and Zhijian Liu. SparseVILA: Decoupling visual sparsity for efficient VLM inference. In *ICCV*, 2025.

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *JMLR*, 9:235–284, 2008.

Ao Li, Yuxiang Duan, Jinghui Zhang, Congbo Ma, Yutong Xie, Gustavo Carneiro, Mohammad Yaqub, and Hu Wang. TransPrune: Token transition pruning for efficient large vision-language model. In *CVPR*, 2026.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. SEED-Bench: Benchmarking multimodal LLMs with generative comprehension. In *CVPR*, 2024.

Yangfu Li, Hongjian Zhan, Tianyi Chen, Qi Liu, Yu jie Xiong, and Yue Lu. Why $1+1<1$ in visual token pruning: Beyond naive integration via multi-objective balanced covering. In *NeurIPS*, 2025.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *EMNLP*, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024a.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, et al. MMBench: Is your multi-modal model an all-around player? In *ICLR*, 2024b.

Yudong Liu, Jingwei Sun, Yueqian Lin, Jianyi Zhang, Jingyang Zhang, Ming Yin, Qinsi Wang, Hai Li, and Yiran Chen. Keyframe-oriented vision token pruning: Enhancing efficiency of large vision language models on long-form video processing. In *ICCV*, 2025.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Aditya Kumar Singh, Hitesh Kandala, Pratik Prabhanjan Brahma, Zicheng Liu, and Emad Barsoum. DUET-VLM: Dual stage unified efficient token reduction for VLM training and inference. In *CVPR*, 2026.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *EMNLP*, 2025.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, et al. PyramidDrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In *CVPR*, 2025.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer is better but not necessary in vision language models. In *CVPR*, 2025.

Hao Yin, Guangzong Si, and Zilei Wang. Lifting the veil on visual information flow in MLLMs: Unlocking pathways to faster inference. In *CVPR*, 2025. URL https://arxiv.org/abs/2503.13108.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in VLMs. In *ICCV*, 2025a.

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in MLLMs. In *NeurIPS*, 2025b.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *ICML*, 2025c.

Xin Zou, Di Lu, Yizhou Wang, Yibo Yan, Yuanhuiyi Lyu, Xu Zheng, Linfeng Zhang, and Xuming Hu. Don't just chase "highlighted tokens" in MLLMs: Revisiting visual holistic context retention. In *NeurIPS*, 2025.

A DERIVATIONS OF THE SELECTION CRITERIA

This section introduces the Gaussian process (GP) model used by *TACI* and derives its two selection criteria. Throughout this section, $\mathbb{S}$ represents a set of selected visual indices and $\mathbb{T}$ denotes the instruction target indices. The M instruction target tokens correspond to the aligned user content and form a subset of the text tokens retained by the decoder. Within each selection stage, the token representations, kernel parameters, and prior covariance matrix are fixed. The posterior covariance is updated after each selection.

A.1 GAUSSIAN PROCESS PRELIMINARIES

A Gaussian process specifies a distribution over functions through a mean function m and a positive semidefinite covariance kernel k (Rasmussen & Williams, 2006):

$$f \sim \mathcal{GP}(m,k), \quad m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})], \quad k(\boldsymbol{x},\boldsymbol{x}') = \mathrm{Cov}(f(\boldsymbol{x}), f(\boldsymbol{x}')). \tag{15}$$

For a finite collection of inputs $X = (\boldsymbol{x}_1,...,\boldsymbol{x}_n)$, this gives

$$\mathbf{f}_X = [f(\boldsymbol{x}_1),...,f(\boldsymbol{x}_n)]^\top \sim \mathcal{N}(\boldsymbol{m}_X, \boldsymbol{K}_{XX}), \quad [\boldsymbol{K}_{XX}]_{i,j} = k(\boldsymbol{x}_i,\boldsymbol{x}_j). \tag{16}$$

In *TACI*, the inputs are token representations and the scalar variables $f(\boldsymbol{x}_i)$ provide an auxiliary model of their dependence. We use a zero mean function, $m(\boldsymbol{x}) = 0$. This does not assume that the token representations or answer distribution are Gaussian.

TACI uses the normalized exponential kernel from Eq. (1). Since $\|\phi(\boldsymbol{x})\| = 1$,

$$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) = \exp\big(\phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j) - 1\big) = \exp\left(-\frac{\|\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_j)\|^2}{2}\right), \tag{17}$$

which is a Gaussian kernel with unit bandwidth. Both selection stages use this kernel, although the representations to which it is applied differ as described in Section C.

Noisy conditioning. Selecting index i is modeled as observing $y_i = f(\boldsymbol{x}_i) + \xi_i$, where $\xi_i \sim \mathcal{N}(0,\varepsilon)$ independently and $\varepsilon > 0$. For a selected set $\mathbb{S}$, standard Gaussian conditioning gives

$$\boldsymbol{C}^{\mathbb{S}} := \mathrm{Cov}(\mathbf{f}_X \mid \mathbf{y}_{\mathbb{S}}) = \boldsymbol{K}_{XX} - \boldsymbol{K}_{X,\mathbb{S}}(\boldsymbol{K}_{\mathbb{S},\mathbb{S}} + \varepsilon \boldsymbol{I})^{-1} \boldsymbol{K}_{\mathbb{S},X}, \tag{18}$$

with $\boldsymbol{C}^{\emptyset} = \boldsymbol{K}_{XX}$. The posterior covariance depends only on the kernel matrix, the observation variance, and the selected indices. *TACI* therefore does not require observed scalar values y_i or ground-truth answers to evaluate its selection criteria.

A.2 DERIVATION OF TARGET-WEIGHTED COVERAGE

Let X contain the N_0 projected visual tokens, let w_u be their normalized relevance weights, and define $\widetilde{w}_u = w_u + \varepsilon_r$. With

$$\boldsymbol{W} = \mathrm{diag}(\widetilde{w}_1,...,\widetilde{w}_{N_0}), \tag{19}$$

the weighted uncertainty objective in Eq. (11) can be written as

$$\mathcal{U}(\mathbb{S}) = \mathrm{tr}(\boldsymbol{W}\boldsymbol{C}^{\mathbb{S}}) = \sum_{u=1}^{N_0} \widetilde{w}_u [\boldsymbol{C}^{\mathbb{S}}]_{u,u}. \tag{20}$$

The sum is taken over all original visual locations, including those that have not been selected. The objective therefore measures how well the selected set represents the weighted visual population.

For an unselected candidate i, conditioning the current GP posterior additionally on its noisy observation gives the rank-one update

$$\boldsymbol{C}^{\mathbb{S}\cup\{i\}} = \boldsymbol{C}^{\mathbb{S}} - \frac{\boldsymbol{C}^{\mathbb{S}}_{:,i} \boldsymbol{C}^{\mathbb{S}}_{i,:}}{[\boldsymbol{C}^{\mathbb{S}}]_{i,i} + \varepsilon}. \tag{21}$$

Taking diagonal entries and substituting them into Eq. (20) gives

$$\begin{aligned} \Delta_{\mathrm{cov}}(i \mid \mathbb{S}) &= \mathcal{U}(\mathbb{S}) - \mathcal{U}(\mathbb{S}\cup\{i\}) \\ &= \sum_{u=1}^{N_0} \widetilde{w}_u \frac{([\boldsymbol{C}^{\mathbb{S}}]_{u,i})^2}{[\boldsymbol{C}^{\mathbb{S}}]_{i,i} + \varepsilon} = \frac{(\boldsymbol{C}^{\mathbb{S}}_{:,i})^\top \boldsymbol{W} \boldsymbol{C}^{\mathbb{S}}_{:,i}}{[\boldsymbol{C}^{\mathbb{S}}]_{i,i} + \varepsilon}. \end{aligned} \tag{22}$$

This is the coverage gain in Eq. (12). The weight is associated with represented location u, rather than directly with candidate i. A candidate can therefore receive a large gain when it reduces uncertainty at several instruction-relevant locations, even if its own relevance weight is modest.

The denominator is positive and the numerator is nonnegative, so each selection weakly decreases $\mathcal{U}$. Summing the gains over a selection order telescopes to the total reduction in weighted uncertainty. This monotonicity does not imply a general submodular approximation guarantee for arbitrary kernels and target weights (Krause et al., 2008).

A.3 Construction and validity of the joint covariance

At the second selection stage, the joint feature set contains the K^0 surviving visual states and the M instruction states. The hidden covariance $\boldsymbol{B}^{\mathrm{hid}}$ is the Gram matrix of the shared kernel κ over this set.

The positive semidefiniteness of κ follows by writing

$$\kappa(\boldsymbol{x},\boldsymbol{y}) = e^{-1}\exp\big(\phi(\boldsymbol{x})^\top \phi(\boldsymbol{y})\big) = e^{-1}\sum_{r=0}^{\infty} \frac{\big(\phi(\boldsymbol{x})^\top \phi(\boldsymbol{y})\big)^r}{r!}. \tag{23}$$

Each term is a positive semidefinite polynomial kernel with a nonnegative coefficient. Hence, κ and its Gram matrix $\boldsymbol{B}^{\mathrm{hid}}$ are positive semidefinite.

For each attention head, the Q/K component is likewise a Gram matrix of κ, evaluated on the visual key and instruction query representations. Each head-wise covariance is therefore positive semidefinite, and their average $\boldsymbol{B}^{\mathrm{QK}}$ is also positive semidefinite.

For $\alpha \in [0,1]$ and any vector $\boldsymbol{a}$,

$$\boldsymbol{a}^\top \boldsymbol{B}\boldsymbol{a} = (1-\alpha)\boldsymbol{a}^\top \boldsymbol{B}^{\mathrm{hid}}\boldsymbol{a} + \alpha \boldsymbol{a}^\top \boldsymbol{B}^{\mathrm{QK}}\boldsymbol{a} \geq 0. \tag{24}$$

Hence the mixed covariance in Eq. (4) is positive semidefinite. Adding observation noise gives

$$\boldsymbol{a}^\top \boldsymbol{\Sigma}\boldsymbol{a} = \boldsymbol{a}^\top \boldsymbol{B}\boldsymbol{a} + \varepsilon\|\boldsymbol{a}\|^2 > 0 \quad (\boldsymbol{a} \neq 0), \quad \boldsymbol{\Sigma} = \boldsymbol{B} + \varepsilon \boldsymbol{I}, \tag{25}$$

so the noisy covariance is positive definite.

A.4 Derivation of Gaussian conditional mutual information

Let $\mathbf{f} \sim \mathcal{N}(0,\boldsymbol{B})$ and $\mathbf{Z} = \mathbf{f} + \boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \mathcal{N}(0,\varepsilon \boldsymbol{I})$. Let $\mathbb{T}$ index the instruction targets and $\mathbb{S}$ the visual indices already selected during Stage 2. From Eq. (18),

$$\boldsymbol{C}^{\mathbb{S}} = \mathrm{Cov}(\mathbf{f} \,|\, \mathbf{Z}_{\mathbb{S}}) = \boldsymbol{B} - \boldsymbol{B}_{:,\mathbb{S}}(\boldsymbol{B}_{\mathbb{S},\mathbb{S}} + \varepsilon \boldsymbol{I})^{-1}\boldsymbol{B}_{\mathbb{S},:}. \tag{26}$$

For a remaining candidate $i \notin \mathbb{S}$, define

$$a_i = [\boldsymbol{C}^{\mathbb{S}}]_{i,i} + \varepsilon, \quad \boldsymbol{c}_i = \boldsymbol{C}^{\mathbb{S}}_{\mathbb{T},i}, \quad \boldsymbol{D}_{\mathbb{T}} = \boldsymbol{C}^{\mathbb{S}}_{\mathbb{T},\mathbb{T}} + \varepsilon \boldsymbol{I}. \tag{27}$$

Conditioned on the selected set, the noisy candidate and instruction variables have covariance

$$\mathrm{Cov}\left(\begin{bmatrix} \mathbf{Z}_i \\ \mathbf{Z}_{\mathbb{T}} \end{bmatrix} \middle| \mathbf{Z}_{\mathbb{S}} \right) = \begin{bmatrix} a_i & \boldsymbol{c}_i^\top \\ \boldsymbol{c}_i & \boldsymbol{D}_{\mathbb{T}} \end{bmatrix}. \tag{28}$$

Conditioning the scalar candidate additionally on the instruction variables gives the Schur complement

$$b_i := \mathrm{Var}(\mathbf{Z}_i \,|\, \mathbf{Z}_{\mathbb{S}}, \mathbf{Z}_{\mathbb{T}}) = a_i - \boldsymbol{c}_i^\top \boldsymbol{D}_{\mathbb{T}}^{-1}\boldsymbol{c}_i. \tag{29}$$

For a scalar Gaussian variable with variance v, the differential entropy is $\frac{1}{2}\log(2\pi e v)$. Therefore,

$$\begin{aligned} I_{\boldsymbol{\Sigma}}(\mathbf{Z}_i; \mathbf{Z}_{\mathbb{T}} \,|\, \mathbf{Z}_{\mathbb{S}}) &= h(\mathbf{Z}_i \,|\, \mathbf{Z}_{\mathbb{S}}) - h(\mathbf{Z}_i \,|\, \mathbf{Z}_{\mathbb{S}}, \mathbf{Z}_{\mathbb{T}}) \\ &= \tfrac{1}{2}\log \frac{a_i}{a_i - \boldsymbol{c}_i^\top \boldsymbol{D}_{\mathbb{T}}^{-1}\boldsymbol{c}_i} \\ &= -\tfrac{1}{2}\log(1-\rho_i^2), \quad \rho_i^2 = \frac{\boldsymbol{c}_i^\top \boldsymbol{D}_{\mathbb{T}}^{-1}\boldsymbol{c}_i}{a_i}. \end{aligned} \tag{30}$$

This recovers Eq. (5). Positive definiteness of the noisy covariance implies $0 \le \rho_i^2 < 1$ in exact arithmetic. In particular, $\boldsymbol{c}_i = 0$ gives zero conditional information gain regardless of the candidate's remaining visual variance.

The same quantity can also be interpreted through the target covariance. After selecting candidate i,

$$\boldsymbol{D}_{\mathbb{T}}^{+} = \boldsymbol{D}_{\mathbb{T}} - \frac{\boldsymbol{c}_i \boldsymbol{c}_i^{\top}}{a_i}. \tag{31}$$

By the matrix determinant lemma, $\frac{1}{2}\log(\det \boldsymbol{D}_{\mathbb{T}} / \det \boldsymbol{D}_{\mathbb{T}}^{+})$ equals the conditional information gain above. The criterion therefore measures reduction in the joint uncertainty of the instruction targets while accounting for correlations among them.

A.5 Residual information and modeling assumptions

Let $\mathbb{V}$ denote the complete Stage 1 candidate pool at the second selection stage. Define

$$\mathcal{R}(\mathbb{S}) = I_{\boldsymbol{\Sigma}}(\mathbf{Z}_{\mathbb{T}}; \mathbf{Z}_{\mathbb{V}\setminus\mathbb{S}} \,|\, \mathbf{Z}_{\mathbb{S}}), \quad F(\mathbb{S}) = I_{\boldsymbol{\Sigma}}(\mathbf{Z}_{\mathbb{T}}; \mathbf{Z}_{\mathbb{S}}). \tag{32}$$

The chain rule gives

$$I_{\boldsymbol{\Sigma}}(\mathbf{Z}_{\mathbb{T}}; \mathbf{Z}_{\mathbb{V}}) = F(\mathbb{S}) + \mathcal{R}(\mathbb{S}), \tag{33}$$

$$F(\mathbb{S}\cup\{i\}) = F(\mathbb{S}) + I_{\boldsymbol{\Sigma}}(\mathbf{Z}_{\mathbb{T}}; \mathbf{Z}_i \,|\, \mathbf{Z}_{\mathbb{S}}). \tag{34}$$

Since $I_{\boldsymbol{\Sigma}}(\mathbf{Z}_{\mathbb{T}}; \mathbf{Z}_{\mathbb{V}})$ is fixed during Stage 2, subtracting the identities before and after adding candidate i gives

$$\mathcal{R}(\mathbb{S}) - \mathcal{R}(\mathbb{S}\cup\{i\}) = I_{\boldsymbol{\Sigma}}(\mathbf{Z}_i; \mathbf{Z}_{\mathbb{T}} \,|\, \mathbf{Z}_{\mathbb{S}}) = \Delta_{\mathrm{CMI}}(i \,|\, \mathbb{S}). \tag{35}$$

Each conditional information step therefore chooses the candidate producing the largest available reduction in residual information. Starting from the empty set, the sum of the gains along any selection order telescopes to $F(\mathbb{S})$.

This incremental procedure does not in general guarantee the globally optimal subset of size K^l. Mutual information with a target is not necessarily submodular without additional assumptions, and conditional gains can increase when variables provide complementary information.

Scope of the surrogate. The preceding identities apply to the fixed Gaussian model used within each selection stage. They characterize information under this surrogate and do not imply preservation of the model's final answer or task performance.

B Computational complexity

Let D be the hidden width, M the number of instruction tokens used by the selection criterion, and M_{all} the number of text tokens in the initial decoder input.

Stage 1. Forming the visual covariance matrix costs $O(N_0^2 D)$, while computing the visual–instruction similarities used for relevance costs $O(N_0 M D)$. Since $q \le 4$, the top-q operation contributes a lower-order term. At each selection step, evaluating the coverage gains for all candidates and updating the posterior covariance both require $O(N_0^2)$ time. Selecting K^0 tokens therefore costs

$$O\big(N_0^2 D + N_0 M D + K^0 N_0^2\big). \tag{36}$$

Stage 2. Let $n = K^0 + M$ denote the number of visual and instruction tokens used to construct the joint covariance. Forming this covariance costs $O(n^2 D)$, and computing the query and key projections costs $O(nD^2)$.

The covariance of the M instruction targets requires an initial $O(M^3)$ Cholesky factorization. After a visual token i is selected, the target covariance changes by the rank-one update

$$\boldsymbol{D}_{\mathbb{T}^l}^{+} = \boldsymbol{D}_{\mathbb{T}^l} - \frac{\boldsymbol{c}_i \boldsymbol{c}_i^{\top}}{[\boldsymbol{C}]_{i,i} + \varepsilon}, \quad \boldsymbol{D}_{\mathbb{T}^l} = \boldsymbol{C}_{\mathbb{T}^l,\mathbb{T}^l} + \varepsilon \boldsymbol{I}. \tag{37}$$

Its Cholesky factor can therefore be maintained in $O(M^2)$ time per selection using a rank-one downdate. Scoring the remaining visual candidates with Cholesky solves costs at most $O(M^2K^0)$ per step, while updating the full posterior covariance costs $O(n^2)$. Stage 2 can therefore be implemented with complexity

$$O\big(n^2D+nD^2+M^3+K^l\big[M^2K^0+n^2\big]\big). \tag{38}$$

The current implementation evaluates the same acquisition criterion but refactorizes the target covariance at each selection step rather than maintaining its Cholesky factor with the rank-one downdate. This affects the measured selection time but is not inherent to the selection criterion.

Decoder computation. We consider the decoder computation excluding the *TACI* selection overhead. Let $l=\lfloor L/2\rfloor$ and $m=\lfloor 3L/4\rfloor$, and define $\mathcal{C}(n)=nD^2+n^2D$, which is the decoder cost of one block for a sequence of length n. After Stage 1, the first l decoder blocks process K^0+M_{all} tokens. After Stage 2, the next $m-l$ blocks process K^l+M_{all} tokens. At layer m, all remaining visual tokens are removed, so the final $L-m$ blocks process only the M_{all} text tokens. The resulting complexity is

$$O\big(l\mathcal{C}(K^0+M_{\text{all}})+(m-l)\mathcal{C}(K^l+M_{\text{all}})+(L-m)\mathcal{C}(M_{\text{all}})\big). \tag{39}$$

Without pruning, the corresponding complexity is

$$O(L\mathcal{C}(N_0+M_{\text{all}})). \tag{40}$$

The FLOPs analysis in Section 4.3 also accounts for visual encoding, token selection, decoder computation, and generation.

C ALGORITHM DETAILS

Algorithm 2 presents the complete inference procedure of *TACI*. We use the kernel in Eq. (1). Throughout the experiments, we set both ε and ε_r to 0.05, and $\alpha=0.5$.

The first selection stage is applied before the decoder blocks using the target-weighted coverage objective. The retained tokens are then propagated to layer $l=\lfloor L/2\rfloor$, where the second selection stage applies the conditional information objective using the mixed covariance in Eq. (4). After each selection stage, the retained visual tokens are restored to their original sequence order before decoding continues. At the final pruning layer $m=\lfloor 3L/4\rfloor$, all remaining visual tokens are removed without further selection.

Both selection stages update the covariance after each selected token using the rank-one posterior update in Algorithm 3, rather than recomputing the kernel. For the conditional-information objective, the instruction covariance is factorized using a numerically stable Cholesky decomposition, and the required quadratic form is computed with Cholesky solves without explicitly forming the matrix inverse.

Algorithm 2 *TACI* Visual token pruning details

Input: visual tokens $\mathbb{V}^0=\{\boldsymbol{v}_i^0\}_{i=1}^{N_0}$, instruction tokens $\mathbb{T}^0=\{\boldsymbol{t}_j^0\}_{j=1}^{M}$
Parameters: budgets K^0,K^l, selection layer $l=\lfloor L/2\rfloor$, final pruning layer $m=\lfloor 3L/4\rfloor$, ε,ε_r,α

Target-weighted visual coverage

Normalize token representations: $\phi(\boldsymbol{x})\leftarrow\boldsymbol{x}/\|\boldsymbol{x}\|_2$
Compute visual–instruction similarities $a_{ij}\leftarrow\phi(\boldsymbol{v}_i^0)^\top\phi(\boldsymbol{t}_j^0)$
$q\leftarrow\min(4,M)$
Compute relevance $r_i\leftarrow\frac{1}{q}\sum_{j\in\mathrm{Top}_q(\{a_{ij}\}_{j=1}^{M})}a_{ij}$ ▷ Eq. (10)
Normalize $w_i\leftarrow\frac{r_i-\min_u r_u}{\max_u r_u-\min_u r_u}$
Construct visual covariance $[\boldsymbol{K}]_{ij}\leftarrow\kappa(\boldsymbol{v}_i^0,\boldsymbol{v}_j^0)$ ▷ Eq. (8)
$\mathbb{S}^0\leftarrow\emptyset$; $\boldsymbol{C}\leftarrow\boldsymbol{K}$
while $|\mathbb{S}^0|<K^0$ **do**
    Compute $g_i^{\mathrm{cov}}\leftarrow\frac{\sum_{u=1}^{N_0}(w_u+\varepsilon_r)[\boldsymbol{C}]_{u,i}^2}{[\boldsymbol{C}]_{i,i}+\varepsilon}$, for each $\boldsymbol{v}_i^0\in\mathbb{V}^0\setminus\mathbb{S}^0$. ▷ Eq. (12)
    $i^*\leftarrow\underset{i:\boldsymbol{v}_i^0\notin\mathbb{S}^0}{\mathrm{argmax}}g_i^{\mathrm{cov}}$
    $\mathbb{S}^0\leftarrow\mathbb{S}^0\cup\{\boldsymbol{v}_{i^*}^0\}$
    $\boldsymbol{C}\leftarrow\textsc{PosteriorUpdate}(\boldsymbol{C},i^*,\varepsilon)$ ▷ Eq. (21) and Algorithm 3
end while
Restore the tokens in $\mathbb{S}^0$ to their original sequence order
Propagate the retained visual tokens and all text tokens to layer l to obtain $\mathbb{V}^l$ and $\mathbb{T}^l$

Conditional information selection

$\mathbb{X}^l\leftarrow\mathbb{V}^l\cup\mathbb{T}^l$
Construct hidden-state covariance ▷ Eq. (2)

$$[\boldsymbol{B}^{\mathrm{hid}}]_{ab}\leftarrow\kappa(\boldsymbol{x}_a^l,\boldsymbol{x}_b^l)$$

Obtain pre-RoPE visual keys $\{\boldsymbol{k}_i^h\}$ and instruction queries $\{\boldsymbol{q}_j^h\}$ at block l
For each head h, associate

$$\boldsymbol{u}_a^h\leftarrow\begin{cases}\boldsymbol{k}_i^h, & \boldsymbol{x}_a^l=\boldsymbol{v}_i^l,\\ \boldsymbol{q}_j^h, & \boldsymbol{x}_a^l=\boldsymbol{t}_j^l\end{cases}$$

Construct query-key covariance ▷ Eq. (3)

$$[\boldsymbol{B}^{\mathrm{QK}}]_{ab}\leftarrow\frac{1}{H}\sum_{h=1}^{H}\kappa(\boldsymbol{u}_a^h,\boldsymbol{u}_b^h)$$

$\boldsymbol{B}\leftarrow(1-\alpha)\boldsymbol{B}^{\mathrm{hid}}+\alpha\boldsymbol{B}^{\mathrm{QK}}$ ▷ Eq. (4)
$\mathbb{S}^l\leftarrow\emptyset$; $\boldsymbol{C}\leftarrow\boldsymbol{B}$
while $|\mathbb{S}^l|<K^l$ **do**
    $\boldsymbol{D}\leftarrow\boldsymbol{C}_{\mathbb{T}^l,\mathbb{T}^l}+\varepsilon\boldsymbol{I}$
    $\boldsymbol{G}\leftarrow\textsc{StableCholesky}(\boldsymbol{D})$
    for all $\boldsymbol{v}_i^l\in\mathbb{V}^l\setminus\mathbb{S}^l$ **do**
        $\boldsymbol{c}_i\leftarrow\boldsymbol{C}_{\mathbb{T}^l,i}$
        $\boldsymbol{x}_i\leftarrow\mathrm{CholeskySolve}(\boldsymbol{G},\boldsymbol{c}_i)$ ▷ Algorithm 3
        $\rho_i^2\leftarrow\frac{\boldsymbol{c}_i^\top\boldsymbol{x}_i}{[\boldsymbol{C}]_{i,i}+\varepsilon}$
        $g_i^{\mathrm{CMI}}\leftarrow-\frac{1}{2}\log(1-\rho_i^2)$ ▷ Eq. (5)
    end for
    $i^*\leftarrow\underset{i:\boldsymbol{v}_i^l\notin\mathbb{S}^l}{\mathrm{argmax}}g_i^{\mathrm{CMI}}$
    $\mathbb{S}^l\leftarrow\mathbb{S}^l\cup\{\boldsymbol{v}_{i^*}^l\}$
    $\boldsymbol{C}\leftarrow\textsc{PosteriorUpdate}(\boldsymbol{C},i^*,\varepsilon)$ ▷ Eq. (21) and Algorithm 3
end while
Restore the tokens in $\mathbb{S}^l$ to their original sequence order
Propagate the retained visual tokens and all text tokens to $m=\lfloor 3L/4\rfloor$
At layer m, remove all remaining visual tokens
Continue decoding with the text tokens

Algorithm 3 Numerically stable covariance operations

function POSTERIORUPDATE($\boldsymbol{C}, i, \varepsilon$)
  $\boldsymbol{c} \leftarrow \mathrm{copy}(\boldsymbol{C}_{:,i})$
  $d \leftarrow \max([\boldsymbol{C}]_{ii}, 0) + \varepsilon$
  $\boldsymbol{C} \leftarrow \boldsymbol{C} - \boldsymbol{c}\boldsymbol{c}^\top / d$
  $\boldsymbol{C} \leftarrow (\boldsymbol{C} + \boldsymbol{C}^\top)/2$
  $[\boldsymbol{C}]_{jj} \leftarrow \max([\boldsymbol{C}]_{jj}, 0)$ for each diagonal entry j
  return $\boldsymbol{C}$
end function

function STABLECHOLESKY($\boldsymbol{D}$)
  for $\delta \in (0, 10^{-12}, 10^{-10}, 10^{-8}, 10^{-6})$ **do**
    Attempt $\boldsymbol{G}\boldsymbol{G}^\top \leftarrow \boldsymbol{D} + \delta \boldsymbol{I}$ using a checked Cholesky factorization
    if the factorization succeeds **then**
      return $\boldsymbol{G}$
    end if
  end for
  Raise a factorization error
end function

function CHOLESKYSOLVE($\boldsymbol{G}, \boldsymbol{b}$)
  Solve $\boldsymbol{G}\boldsymbol{y} = \boldsymbol{b}$ for $\boldsymbol{y}$
  Solve $\boldsymbol{G}^\top \boldsymbol{x} = \boldsymbol{y}$ for $\boldsymbol{x}$
  return $\boldsymbol{x}$
end function

D EXPERIMENTAL DETAILS

D.1 TOKEN BUDGETS AND REDUCTION RATIOS

We evaluate three token-reduction settings. Since the baselines prune visual tokens at different locations and stages, we characterize the computational budget by the average number of visual tokens processed across the LLM layers:

$$\text{Average tokens} = \frac{1}{L}\sum_{\ell=0}^{L-1} n_\ell, \tag{41}$$

where n_ℓ is the number of visual tokens processed by decoder block ℓ.

Token budgets for *TACI*. *TACI* operates at three pruning locations, with their positions adjusted to the depth of each backbone. Using zero-based layer indices, the pruning locations are

$$\mathcal{L}_p = \begin{cases} \{0,16,24\}, & \text{LLaVA-1.5,} \\ \{0,14,21\}, & \text{Qwen2.5-VL.} \end{cases} \tag{42}$$

Here, $\ell = 0$ corresponds to the first selection stage before the first LLM layer. The second selection stage is placed halfway through the decoder, and all remaining visual tokens are removed at the final pruning location after approximately three quarters of the LLM layers.

Token-budget matching for baselines. For each baseline, we adjust the number of retained visual tokens at its original pruning stages to match the average token budgets in Eq. (41) as closely as possible, while keeping the original pruning locations and number of pruning stages fixed.

For Qwen2.5-VL, dynamic-resolution image encoding makes the initial number of visual tokens vary across examples. This particularly affects PyramidDrop and TransPrune, which begin pruning only after several LLM layers. Since all visual tokens are processed before the first pruning stage, sufficiently small average token budgets cannot be reached without changing the original pruning schedule. Reaching such budgets would require additional design choices, e.g. moving the pruning locations or adding pruning stages, so we preserve the original pruning schedules in our evaluation.

For example, Qwen2.5-VL contains 28 LLM layers. Consider an input with 576 visual tokens and a method whose first pruning stage occurs after the first seven layers. These layers alone process $7 \times 576 = 4{,}032$ visual tokens, which already exceeds the total layer-wise budget of $28 \times 128 = 3{,}584$ corresponding to an average of 128 tokens. When the requested budget is below the minimum achievable budget, we use the most aggressive feasible configuration without changing the original pruning schedule. Consequently, the 64- and 128-token targets can resolve to the same pruning configuration.

When the two targets produce the same configuration for all examples in a benchmark, the 64-token result would duplicate the 128-token result and is therefore not presented separately in Table 2.

D.2 DATASETS

We evaluate our method on seven benchmarks spanning diverse aspects of visual understanding and reasoning.

GQA (Hudson & Manning, 2019) evaluates compositional visual reasoning using questions generated from scene graphs in Visual Genome. The questions cover objects, attributes, relations, and multi-step combinations of these properties. Our evaluation set contains $12{,}578$ questions, and we report exact-match accuracy between the predicted and ground-truth answers.

MME (Fu et al., 2025) evaluates 14 perception and cognition tasks, including object existence, counting, position, color, text recognition, and commonsense reasoning, using yes–no questions. Each image is paired with two questions, one answered *yes* and the other *no*, allowing the benchmark to evaluate both individual question accuracy and image-level consistency.

For category c, let a_c denote question-level accuracy and p_c the fraction of images for which both paired questions are answered correctly. The category score is

$$100(a_c + p_c),$$

and the final MME score sums this value over all 14 categories, with a maximum score of $2{,}800$. The $2{,}374$ questions in our evaluation preserve the original image-level pairs.

POPE (Li et al., 2023) probes object hallucination using questions about whether a named object occurs in an image. Its standard COCO construction samples absent objects randomly, by frequency, or by co-occurrence to create different negative-question regimes. The prepared set contains $9{,}000$ records. Accuracy uses the leading yes or no.

MMBench (Liu et al., 2024b) evaluates a broad range of visual perception and reasoning abilities using multiple-choice questions collected from Internet sources and public datasets. To reduce sensitivity to the ordering of answer choices, MMBench adopts circular evaluation, where the options of each question are permuted to form multiple variants. Our prepared evaluation set contains $4{,}329$ records, including these variants. We group variants belonging to the same original question and count the question as correct only when all of its variants are answered correctly. We report the resulting circular-group accuracy.

MMStar (Chen et al., 2024b) evaluates six core multimodal capabilities—coarse perception, fine-grained perception, instance reasoning, logical reasoning, science and technology, and mathematics—using $1{,}500$ multiple-choice questions. Its questions are curated to require visual input and to reduce the effects of data leakage, providing a more direct measure of vision-dependent multimodal reasoning. We report multiple-choice accuracy based on the parsed answer option.

VizWiz (Gurari et al., 2018) consists of photographs and spoken questions submitted by blind users in real-world settings. The dataset contains challenging images with blur, poor framing, or insufficient visual evidence, and some questions are explicitly unanswerable. Each question is annotated with 10 crowdsourced answers, reflecting the ambiguity that can arise in these real-world images. The prepared validation set contains $4{,}319$ question–image pairs.

We follow the standard VQA consensus metric. For a predicted answer a, the score increases with the number of human annotators who provide the same answer and reaches full credit when at least three annotators agree with the prediction. Thus, unlike exact-match accuracy against a single reference, this metric accounts for agreement among multiple valid human answers. We report the average VQA accuracy over all questions.

SEED-Bench (Li et al., 2024) evaluates multimodal comprehension using four-choice multiple-choice questions, with one correct answer and three distractors for each question. Its image-based evaluation covers a range of visual understanding abilities, including scene understanding, object identity and attributes, spatial relations, instance counting, and visual reasoning. Questions are generated from visual information extracted from the images and are subsequently filtered automatically and verified by human annotators. Our prepared image evaluation contains $14{,}232$ questions. We report the percentage of questions for which the predicted option matches the ground-truth option.

D.3 Prompts

The prompts used for evaluation consist of fixed instructions and example-specific components, as shown in Figure 3. The variable components include the question and, for multiple-choice benchmarks, the answer options and optional hint. We use short-answer prompts for GQA and VizWiz, option-selection prompts for MMBench, MMStar, and SEED-Bench, and yes or no prompts for MME and POPE. The textual content is kept fixed across backbones, while each model uses its native multimodal input format and chat template.

D.4 Implementation details

Backbones and inference setup. We evaluate *TACI* on LLaVA-1.5-7B[2] and Qwen2.5-VL-7B-Instruct.[3] All model parameters remain fixed, and all experiments are conducted on NVIDIA RTX 6000 Ada GPUs. For answer generation, we use greedy decoding.

Feature extraction. Stage 1 uses the visual embeddings produced by the native projector or visual merger and the input embeddings of the instruction tokens. Stage 2 uses the surviving visual hidden

[2] https://huggingface.co/liuhaotian/llava-v1.5-7b

[3] https://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct

GQA prompt

```
[Image tokens]
{question}
Answer the question using a single word or phrase.
```

VizWiz prompt

```
[Image tokens]
{question}
When the
provided information is insufficient, respond with 'Unanswerable'.
Answer the question using a single word or phrase.
```

MMBench / MMStar / SEED-Bench prompt

```
[Image tokens]
Hint: {hint}
Question: {question}
Options:
{options}
Please select the correct answer from the options above.
Answer with
only the letter of the correct option. Do not include an explanation.
```

MME prompt

```
[Image tokens]
{question} Please answer yes or no.
```

POPE prompt

```
[Image tokens]
{question} Please answer yes or no.
```

Figure 3: Prompt templates used for the seven evaluation benchmarks. Variable components are indicated by placeholders in braces. GQA and VizWiz use short-answer prompts. MMBench, MMStar, and SEED-Bench use multiple-choice prompts, while MME and POPE use yes or no prompts. `[Image tokens]` schematically denotes the visual input and is not part of the textual prompt. Each backbone uses its native multimodal input format and chat template.

states and the corresponding contextualized instruction representations. Text tokens outside the instruction target remain in the decoder sequence but are not used as selection targets.

At Stage 2, query and key features are obtained using the input normalization and query and key projections of the following decoder block. Any native normalization of the query and key features is retained, and key heads are expanded to match the grouped-query attention layout when necessary. The kernel over the query and key features is computed before rotary positional embeddings and without applying the attention softmax.

Sequence handling. After each selection stage, the retained visual tokens are restored to their original sequence order. Visual hidden states, positional information, attention masks, and KV-cache entries are compacted using the same selected indices. At the final pruning layer, all remaining visual tokens are removed while the text tokens are retained.

Table 3: Component ablations on LLaVA-1.5-7B with an average visual-token budget of 64, evaluated on subsets of GQA, POPE, and MMStar. Panel (a) fixes Stage 2 to CMI with the mixed covariance, while panels (b) and (c) fix Stage 1 to target-weighted coverage. The hidden-only and mixed covariance settings use $\alpha = 0$ and $\alpha = 0.5$, respectively. All datasets are reported as accuracy (%).

Selection variant	GQA	POPE	MMStar
(a) Stage 1: relevance weighting			
Uniform coverage	49.02	85.55	52.73
Instruction-relevance ranking	46.29	82.42	50.39
Target-weighted coverage (Ours)	**52.54**	**88.09**	**56.05**
(b) Stage 2: conditioning and coverage control			
Target-weighted coverage (visual covariance)	51.37	86.33	53.32
Static MI (mixed covariance)	50.59	79.88	54.30
Conditional MI (mixed covariance, Ours)	**52.54**	**88.09**	**56.05**
(c) Stage 2: covariance construction			
CMI with hidden covariance	51.95	85.74	55.08
CMI with mixed covariance (Ours)	**52.54**	**88.09**	**56.05**

E Ablation studies

Stage 1 of *TACI* must preserve an effective candidate pool, while Stage 2 must select complementary evidence from that pool. We ablate three design choices: How instruction relevance is incorporated in Stage 1, whether Stage 2 conditions on previously selected tokens, and how the Stage 2 covariance is constructed. Table 3 reports the results for these design choices.

Stage 1 relevance weighting. Stage 2 can only select tokens that survive Stage 1. Table 3(a) compares three ways to construct this candidate pool. Uniform coverage gives every visual location equal priority. Instruction-relevance ranking selects the most relevant tokens individually. Target-weighted coverage instead gives higher priority to representing relevant locations, allowing a token to be useful because of the content it covers.

Stage 2 conditioning. Individually relevant tokens can carry overlapping evidence. Table 3(b) compares static mutual information (MI) with conditional MI (CMI). Static MI scores each candidate once by its mutual information with the instruction, without conditioning on previously selected visual tokens. CMI instead updates the score after each selection by conditioning on the selected set, thereby measuring the additional information provided by each candidate. Using the same mixed covariance isolates the effect of conditioning on the selected set. We also replace Stage 2 with target-weighted coverage to compare *TACI* with coverage at both stages. This control changes both the selection objective and the covariance construction.

Stage 2 covariance construction. Table 3(c) keeps the CMI selection rule fixed and compares the hidden covariance with the proposed mixed covariance.

F Additional examples

We provide additional qualitative examples comparing the visual tokens retained by *TACI* and the baseline pruning methods Figures 4 and 5.

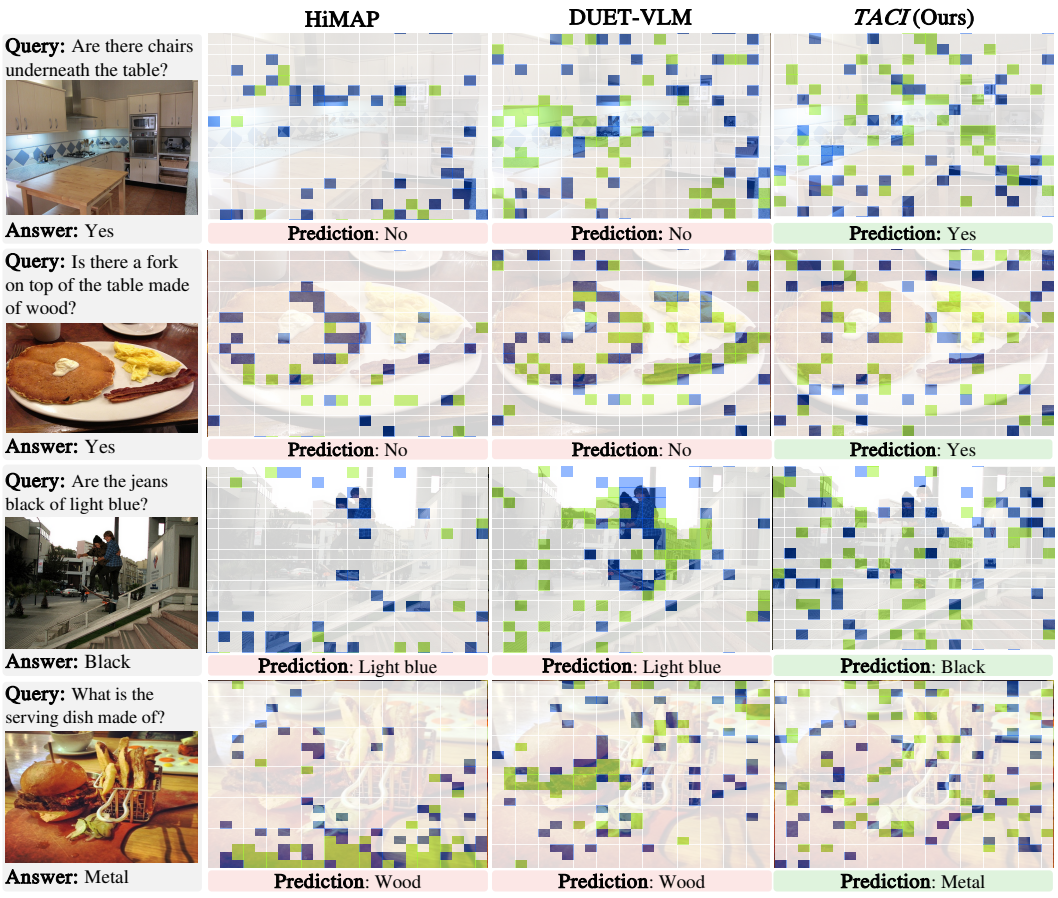

Figure 4: Examples of visual evidence retained by different token pruning methods on LLaVA-1.5 for MME. For *TACI*, tokens selected by target-weighted coverage are highlighted in green and blue. The tokens retained after conditional information selection are highlighted in blue.

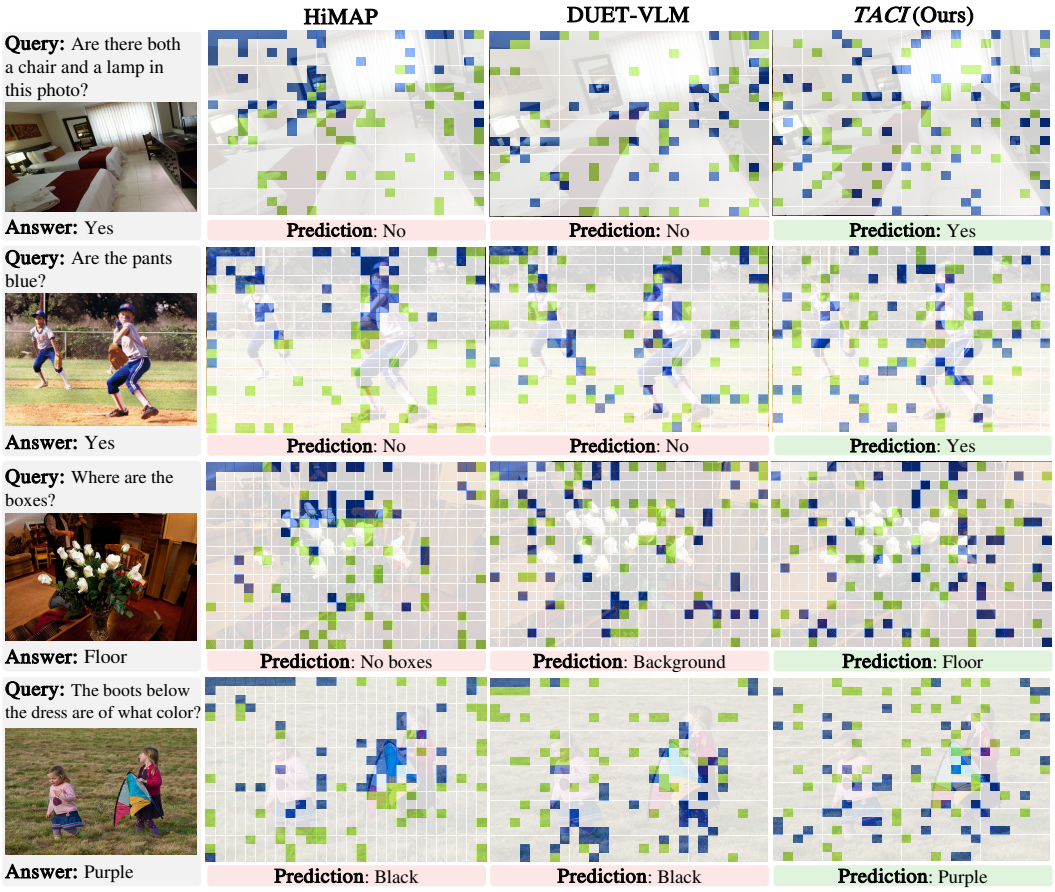

Figure 5: Examples of visual evidence retained by different token pruning methods on Qwen2.5-VL for MME. For *TACI*, tokens selected by target-weighted coverage are highlighted in green and blue. The tokens retained after conditional information selection are highlighted in blue.