FOCUS ON WHAT MATTERS: TEXT-AWARE, TRAINING-FREE TOKEN PRUNING FOR EFFICIENT DOCUMENT PARSING

Anonymous authors
Paper under double-blind review

ABSTRACT

Document parsing methods based on vision-language models (VLMs) typically encode high-resolution document images into dense sequences of visual tokens, resulting in high computational costs. Many of these tokens correspond to blank or background regions, resulting in redundant computation. Token pruning offers an effective way to alleviate this overhead by reducing the number of visual tokens. However, general-purpose token pruning methods suffer from two main limitations: **(1)** they typically rank visual tokens based on attention scores or feature similarity, potentially discarding tokens from critical text regions in the image; **(2)** they struggle to adapt pruning ratios to varying text densities. Both limitations degrade parsing accuracy. To address these limitations, we propose **T^3-Doc**, a Text-aware, Training-free Token pruning framework for efficient Document parsing. T^3-Doc employs an off-the-shelf text detector to locate text regions and maps them onto the VLM's token grid to construct a retention mask. The framework incorporates two key designs: **(1)** Flexible Pruning Placement, supporting pruning before or after the visual encoder, with pre-encoder pruning further reducing the encoder's computational cost; **(2)** Coverage-Adaptive Thresholding, which uses region dilation and bounding-box merging to regulate mask coverage, thereby adaptively adjusting the token retention ratio. Extensive experiments demonstrate that T^3-Doc prunes approximately 30% of visual tokens on average while largely maintaining parsing accuracy, achieving a better accuracy–efficiency trade-off than existing methods. The code will be publicly available.

1 INTRODUCTION

Recent advances in vision-language models (VLMs) have enabled end-to-end document parsing that jointly handles text, formulas, tables, and complex page layouts (Wei et al., 2024; Poznanski et al., 2025; Li et al., 2025; Team et al., 2025). These models improve the completeness and consistency of page-level parsing but also introduce substantial computational overhead (Li et al., 2025; Team et al., 2025; Niu et al., 2026), as they typically encode high-resolution document images into dense visual token sequences (Wei et al., 2025; Wang et al., 2026). However, document pages often contain blank or background regions, such as margins and gaps between lines or columns, that contain little or no textual content (Liu et al., 2026; Wang et al., 2026). Because standard image tokenization divides the entire image into patches, these regions are also converted into visual tokens and processed alongside text-bearing tokens, leading to redundant computation (Dosovitskiy et al., 2021; Wang et al., 2024b). Token pruning offers an effective way to alleviate this computational overhead by reducing the number of visual tokens.

However, general-purpose token pruning methods face two crucial limitations in document parsing. First, these methods typically rank visual tokens based on attention scores or feature similarity, but these criteria do not directly reflect the importance of tokens for text recognition (Chen et al., 2024; Zhang et al., 2025b;a; Xing et al., 2025; Shang et al., 2025). Even a character or digit occupying a small region can be essential for correct parsing, so discarding low-scoring tokens may remove critical text content (Liu et al., 2026; Ding et al., 2026). As illustrated in Figure 1, generic pruning may remove tokens from table headers and numerical entries, leading to parsing errors. Second, these methods typically apply a fixed token retention ratio to pages with varying text densities,

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.

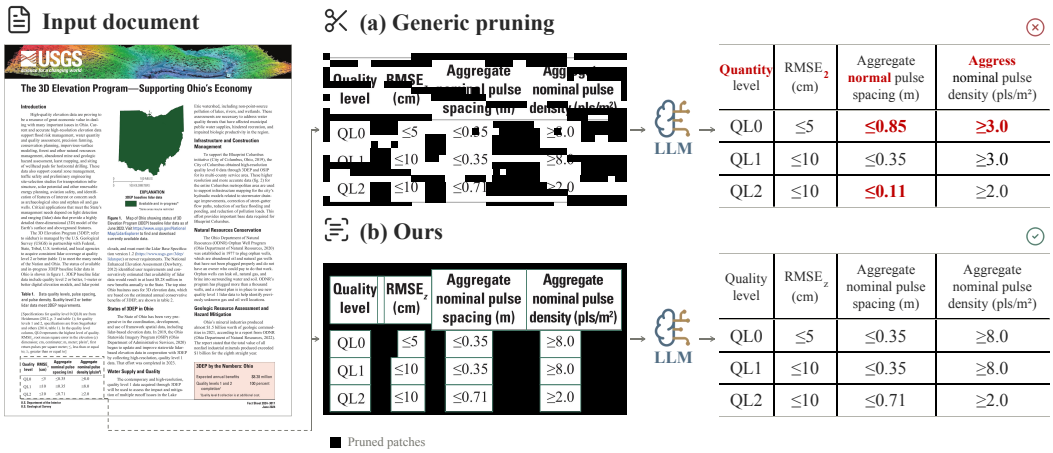


Figure 1: Comparison of generic token pruning and T^3-Doc on the same document page. Black patches indicate pruned regions. (a) Generic pruning based on model-derived scores removes patches containing parts of table headers and numerical entries, leading to parsing errors (shown in red). (b) T^3-Doc uses detected text regions to guide token retention, preserving the table content needed for correct parsing in this example.

discarding critical content from text-dense pages while retaining too many redundant tokens on text-sparse pages. These limitations highlight the need for document-specific token pruning that preserves critical text regions and adapts to varying text densities.

To address these limitations, we propose T^3-Doc, a Text-aware, Training-free Token pruning framework for efficient Document parsing. T^3-Doc employs an off-the-shelf text detector to locate text regions and maps them onto the VLM's token grid to construct a retention mask. This framework requires no additional training or modifications to the VLM's weights. This text-guided selection aims to preserve visual tokens corresponding to critical text regions while reducing redundant computation in blank and background regions. The framework incorporates two key designs: (1) Flexible Pruning Placement, supporting pruning before or after the visual encoder, with pre-encoder pruning further reducing the encoder's computational cost; (2) Coverage-Adaptive Thresholding, which uses region dilation and bounding-box merging to regulate mask coverage, thereby adaptively adjusting the token retention ratio.

The effectiveness of T^3-Doc is verified through extensive experiments. Experimental results demonstrate that T^3-Doc prunes approximately 30% of visual tokens on average while largely maintaining parsing accuracy, achieving a better accuracy–efficiency trade-off than existing token pruning methods. Moreover, T^3-Doc is compatible with existing acceleration techniques such as FlashAttention (Dao et al., 2022), further highlighting its potential for practical deployment.

Our main contributions are summarized as follows:

- We propose T^3-Doc, a Text-aware, Training-free visual Token pruning framework for efficient Document parsing. The framework employs an off-the-shelf text detector to locate text regions and maps them onto the VLM's token grid to construct a retention mask, reducing redundant computation in background regions.
- We introduce two key designs: (1) Flexible Pruning Placement, which supports pruning before or after the visual encoder, with pre-encoder pruning further reducing the encoder's computational cost; (2) Coverage-Adaptive Thresholding, which uses region dilation and bounding-box merging to regulate mask coverage, thereby adaptively adjusting the token retention ratio.
- Experimental results demonstrate that T^3-Doc prunes approximately 30% of visual tokens on average while largely maintaining parsing accuracy, achieving a better accuracy–efficiency trade-off than existing methods. Furthermore, the pruning mechanism of T^3-Doc is orthogonal to existing acceleration frameworks.

2 Related Work

Document Parsing based on Vision-Language Models. Recent VLMs, including Qwen3-VL (Bai et al., 2025), GOT (Wei et al., 2024), olmOCR (Poznanski et al., 2025), and dots.ocr (Li et al., 2025), combine visual perception with language generation to jointly parse text, formulas, tables, and page layouts. To handle high-resolution documents, existing methods explore dynamic-resolution encoding, region-based processing, and visual compression. For example, HunyuanOCR (Team et al., 2025) adopts dynamic-resolution encoding to accommodate document images of varying sizes. PaddleOCR-VL (Cui et al., 2026) and MinerU2.5 (Niu et al., 2026) combine layout detection with region recognition, while DeepSeek-OCR (Wei et al., 2025) uses visual compression to reduce the number of tokens required to represent a document. Despite these advances in parsing capability and efficiency, processing visual tokens from high-resolution documents remains computationally expensive. To reduce this cost, document-specific acceleration methods such as LayoutLite (Liu et al., 2026) and FastOCR (Tang et al., 2026) employ learned token selection and sparse attention, respectively. In contrast, T^3-Doc uses explicit text regions to guide pruning without additional training, and its pruning mechanism is orthogonal to existing acceleration techniques such as FlashAttention (Dao et al., 2022).

Visual-Token Pruning. Existing visual token reduction methods for general-purpose VLMs select or merge tokens based on attention scores, feature similarity, or other criteria. Within the language model, FastV (Chen et al., 2024) prunes visual tokens using early-layer attention, SparseVLM (Zhang et al., 2025b) uses text-guided visual attention, PyramidDrop (Xing et al., 2025) adopts a stage-wise reduction schedule, and AVTP (Zhang et al., 2026a) selects pruning layers based on attention statistics. V2Drop (Chen et al., 2025) instead identifies redundant tokens through changes in hidden states across layers. Among methods based on visual encoder features, FasterVLM (Zhang et al., 2025a) ranks tokens using class-token attention, PruMerge (Shang et al., 2025) and VisionZip (Yang et al., 2025) combine token selection with merging, and DivPrune (Alvar et al., 2025) selects a subset of visual tokens based on feature diversity. PixelPrune (Wang et al., 2026) performs pruning before the visual encoder, removing patches based on pixel redundancy. Although these methods reduce the number of visual tokens, their selection criteria do not directly reflect the importance of text regions in documents and may therefore discard characters or digits essential for parsing. Moreover, a fixed token retention ratio does not account for differences in text density across pages, limiting the ability of these methods to balance content preservation and redundancy reduction. In contrast, T^3-Doc constructs a retention mask from detected text regions and uses Coverage-Adaptive Thresholding to regulate mask coverage, allowing the token retention ratio to adapt to each page's text distribution.

3 Method

T^3-Doc is a text-aware, training-free token pruning framework for document parsing. It constructs a retention mask by mapping detected text regions onto the VLM's token grid, without relying on model-derived importance scores or modifying the VLM's weights (Section 3.2). The framework incorporates two key designs: (1) *Flexible Pruning Placement* supports pruning before or after the visual encoder, with pre-encoder pruning further reducing the encoder's computational cost (Section 3.3). (2) *Coverage-Adaptive Thresholding* uses bounding-box merging and region dilation to regulate mask coverage, thereby adaptively adjusting the token retention ratio (Section 3.4). Figure 2 illustrates the overall framework.

3.1 Problem Setup

A typical document VLM processes an input image through a visual encoder and a projection or token-merging module to obtain visual tokens for an autoregressive language model. Given a document image I, visual encoding and merging produce a spatial grid of visual tokens with dimensions $H_g \times W_g$:

$$X = \mathrm{vec}\left(\{x_{u,v}\}_{u=1,\ldots,H_g;\, v=1,\ldots,W_g}\right) = (x_1, \ldots, x_N), \qquad N = H_g W_g, \tag{1}$$

where $\mathrm{vec}(\cdot)$ follows the VLM's native visual-token sequence order.

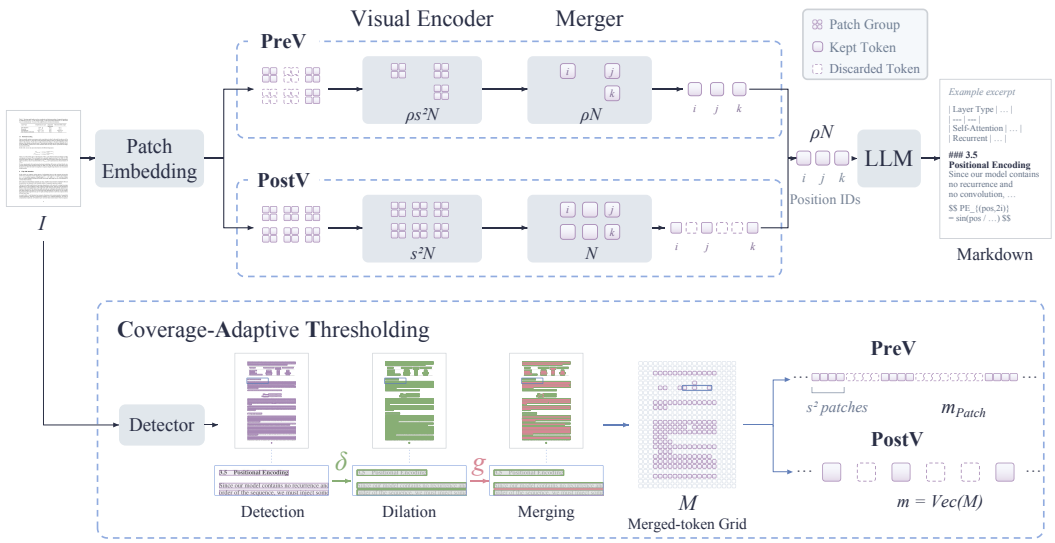


Figure 2: Overview of T^3-Doc. *Bottom:* Coverage-Adaptive Thresholding (CAT) converts detected text regions into a merged-token retention mask M through bounding-box merging (g) and region dilation (δ). *Top:* T^3-Doc-PreV applies the mask before the visual encoder, while T^3-Doc-PostV applies it after the merger. Both variants retain ρN visual tokens for the language model while preserving their original position IDs.

These visual tokens are processed by the language model together with the text prompt tokens to generate a structured document representation. Token pruning aims to improve computational efficiency by removing redundant visual tokens. A pruning decision is represented by a binary retention mask $m \in \{0,1\}^N$, where $m_i = 1$ indicates that the i-th visual token is retained. The retained tokens form a subsequence X_m in their original sequence order, with a retention ratio defined as

$$\rho = \frac{|X_m|}{N} = \frac{1}{N}\sum_{i=1}^{N} m_i. \tag{2}$$

3.2 Text-Aware Retention Mask

The text-aware retention mask is constructed from detected text regions. Given a document image I of width W_0 and height H_0, an off-the-shelf text detector identifies K text regions, each represented by an axis-aligned bounding box in the original image coordinates:

$$\mathcal{B}(I) = \{b_j\}_{j=1}^{K}, \qquad b_j = (x_j^{\min}, y_j^{\min}, x_j^{\max}, y_j^{\max}). \tag{3}$$

Polygon-shaped detections are converted to their enclosing axis-aligned bounding boxes. Coverage-Adaptive Thresholding adjusts the detected regions through bounding-box merging and region dilation, yielding the adjusted region set $\mathcal{Q}_{\delta,g}$ (Section 3.4). The adjusted regions are mapped onto the VLM's token grid by scaling each point (x, y) in the original image coordinates to its corresponding grid coordinates $(\tilde{x}, \tilde{y})$, which gives the grid-space region set $\tilde{\mathcal{Q}}_{\delta,g}$:

$$\tilde{x} = x\frac{W_g}{W_0}, \qquad \tilde{y} = y\frac{H_g}{H_0}. \tag{4}$$

Let $C_{u,v}$ denote the spatial extent of grid cell (u, v). The binary retention mask is defined as

$$M_{u,v} = \begin{cases} 1, & \exists\, \tilde{q} \in \tilde{\mathcal{Q}}_{\delta,g} :\ C_{u,v} \cap \tilde{q} \neq \varnothing, \\ 0, & \text{otherwise}, \end{cases} \qquad m = \mathrm{vec}(M). \tag{5}$$

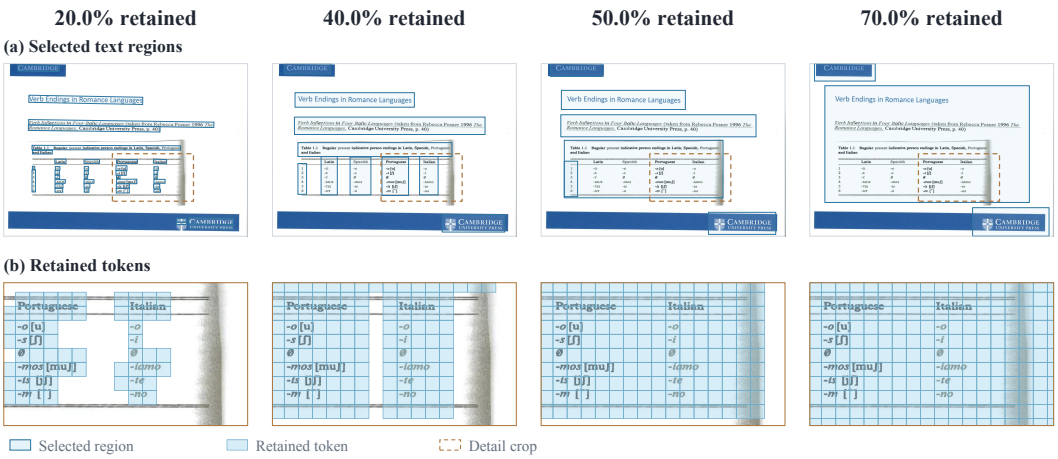


Figure 3: Visualization of **C**overage-**A**daptive **T**hresholding under different budgets. (a) Selected text regions under different retention budgets. (b) Corresponding retained visual tokens in the highlighted region, with progressively broader contextual coverage as more tokens are retained.

3.3 Flexible Pruning Placement

The text-aware retention mask can be applied prior to or following visual encoding, as illustrated in the top half of Figure 2. The corresponding variants are denoted as T^3-Doc-PreV and T^3-Doc-PostV, respectively. T^3-Doc-PostV applies the mask directly to the visual tokens generated by the visual encoder, whereas T^3-Doc-PreV uses it to select image patches before visual encoding, further reducing the encoder's computational cost.

T^3-Doc-PostV. The document image is processed by the visual encoder and token merger to produce N visual tokens. The retention mask m is then applied to obtain X_m, and only the retained ρN visual tokens are passed to the language model. Given a text prompt of length T, the language model's input length decreases from $N + T$ to $\rho N + T$, reducing prefill computation and the KV cache associated with visual tokens.

T^3-Doc-PreV. Token pruning is performed after patch embedding and before visual encoding. Let s denote the spatial merge size, so that each merged visual token corresponds to an $s \times s$ group of image patches. The retention mask m is expanded to the patch level by repeating each mask value s^2 times:

$$m_{\text{patch},k} = m_{\lceil k/s^2 \rceil}, \qquad k = 1, \ldots, s^2 N, \tag{6}$$

where the s^2 patch tokens within each merge window are contiguous in the processor's sequence order. Patch tokens with a mask value of 0 are removed. The remaining $\rho s^2 N$ patch tokens are processed by the visual encoder and merged into ρN visual tokens for the language model. Each retained patch token preserves its original positional encoding.

3.4 Coverage-Adaptive Thresholding

Coverage-Adaptive Thresholding uses bounding-box merging and region dilation to adjust mask coverage, preserving local context that tight detection boxes may omit while adaptively regulating the token retention ratio, as illustrated in Figure 3. Bounding-box merging is applied first, and region dilation then acts on the merged regions.

Bounding-Box Merging. Let $U = \bigcup_{j=1}^{K} b_j$ denote the union of the original detection boxes. Morphological dilation with radius g is applied to U to connect regions separated by at most $2g$. Each connected component of the dilated region is then enclosed by an axis-aligned bounding box, yielding

$$\mathcal{Q}_g = \{\text{box}(A) : \ A \in \text{cc}(U \oplus D_g)\}, \qquad g > 0, \tag{7}$$

Table 1: Comparison with token-pruning methods on OmniDocBench. All methods use the same backbone and retain 70% of the merged visual tokens ($\rho = 0.70$); w/o pruning keeps all tokens. Among pruning methods, the best result in each column is in **bold** and the second best is underlined.

Method	**Overall**↑	**Text**$^{\text{Edit}}$↓	**Formula**$^{\text{CDM}}$↑	**Table**$^{\text{TEDS}}$↑	**Table**$^{\text{TEDS-S}}$↑	**Reading Order**$^{\text{Edit}}$↓
w/o pruning	92.65	0.0407	93.77	88.24	91.27	0.1348
Random (PostV)	84.38	0.1581	89.98	78.97	84.25	0.2014
Random (PreV)	61.48	0.3332	58.45	59.32	71.58	0.2818
DivPrune (Alvar et al., 2025)	88.45	0.0900	92.44	81.92	86.88	0.1671
FasterVLM (Zhang et al., 2025a)	90.93	0.0765	94.06	86.39	90.43	0.1578
FastV (Chen et al., 2024)	91.04	0.0683	93.63	86.31	89.98	0.1516
PixelPrune (Wang et al., 2026)	88.98	0.0932	92.66	83.61	87.73	0.1681
PruMerge (Shang et al., 2025)	91.58	0.0671	94.88	86.56	90.42	0.1514
SparseVLM (Zhang et al., 2025b)	82.25	0.1822	88.17	76.79	82.51	0.2099
V2Drop (Chen et al., 2025)	89.44	0.0826	92.72	83.85	88.86	0.1573
T^3-Doc-PreV (**Ours**)	92.02	**0.0392**	93.89	86.07	89.44	0.1333
T^3-Doc-PostV (**Ours**)	**92.81**	0.0416	**94.94**	**87.63**	**90.74**	**0.1317**

where D_g is a disk of radius g, $\oplus$ denotes morphological dilation, cc extracts connected components, and $\mathrm{box}(A)$ returns the axis-aligned bounding box of component A. This operation combines nearby text boxes into larger regions, covering gaps between text lines or blocks. When $g = 0$, the original detection boxes are retained without merging, i.e., $\mathcal{Q}_0 = \mathcal{B}(I)$.

Region Dilation. Each region $q \in \mathcal{Q}_g$ is expanded by moving each vertex v outward by δ pixels relative to its center c_q:

$$v' = v + \delta \frac{v - c_q}{\|v - c_q\|_2}. \tag{8}$$

The updated vertices v' are connected in their original order to form an expanded region q' that includes additional context around text boundaries, giving $\mathcal{Q}_{\delta,g} = \{q' : q \in \mathcal{Q}_g\}$. These regions are mapped onto the token grid and combined into the retention mask m following Section 3.2.

4 Experiments

4.1 Implementation Details

The experiments are mainly conducted on OmniDocBench (Ouyang et al., 2025) using the aggregate Overall score and element-specific metrics. Specifically, normalized Edit Distance is used for text recognition and reading order, CDM (Wang et al., 2024a) for formula recognition, and TEDS together with its structure-only variant TEDS-S (Zhong et al., 2020) for table parsing. The main experiments are conducted based on FireRed-OCR (Wu et al., 2026). All methods share the same model checkpoint, image processor, prompt template, input image resolution, and run in `bfloat16` precision with a batch size of 1. Unless otherwise specified, all methods use greedy decoding with a maximum output length of 8,192 tokens. All inference and profiling experiments are conducted on a single NVIDIA H800 GPU, with PyTorch SDPA as the default attention backend. Text regions are extracted using the PP-OCRv6-Medium detector (Zhang et al., 2026b). Unless otherwise specified, CAT targets a mean token retention ratio of 70%, while each page's actual retention ratio is determined by its mask coverage. Random pruning matches the per-page retained token counts of T^3-Doc at the corresponding pruning placement. All compared methods are re-implemented using the same FireRed-OCR checkpoint, with a default visual token retention ratio of 70%.

4.2 Main Results

Comparison with token-pruning methods. We compare T^3-Doc with representative token-pruning methods on OmniDocBench at a target mean token retention ratio of 70%. The results are presented in Table 1. Based on these results, we draw the following key conclusions. First, T^3-Doc achieves a favorable balance between token reduction and parsing accuracy, with both variants outperforming all compared token-pruning methods in Overall score. Specifically, T^3-Doc-PostV achieves an Overall score of 92.81 while removing approximately 30% of visual tokens, exceeding

Table 2: Generalization of T^3-Doc across document VLMs on OmniDocBench (Ouyang et al., 2025), including OvisOCR2 (Lu et al., 2026), WeVisDoc-4B (Yu et al., 2026), and Logics-Parsing-v2 (Logics-MLLM Team, 2026). All backbones use the same text detector and mask construction procedure. ρ denotes the token retention ratio, with $\rho = 1.00$ corresponding to the unpruned baseline. PostV and PreV denote pruning after and before the visual encoder, respectively. The best result in each metric for each backbone is highlighted in **bold**.

Model	Stage	ρ	**Overall**↑	**Text**$^{\text{Edit}}$↓	**Formula**$^{\text{CDM}}$↑	**Table**$^{\text{TEDS}}$↑	**Table**$^{\text{TEDS-S}}$↑	**Reading Order**$^{\text{Edit}}$↓
OvisOCR2	*w/o pruning*	1.00	**95.89**	**0.0308**	**96.75**	**94.01**	**96.35**	**0.1155**
	PostV	0.70	95.41	0.0346	96.66	93.02	95.65	0.1207
	PreV	0.70	95.15	0.0324	96.25	92.44	95.11	0.1206
WeVisDoc-4B	*w/o pruning*	1.00	**94.87**	0.0411	**96.54**	**92.17**	94.47	0.1297
	PostV	0.70	94.64	**0.0393**	95.76	92.10	**94.53**	**0.1277**
	PreV	0.70	94.50	0.0408	95.90	91.67	94.07	0.1311
Logics-Parsing-v2	*w/o pruning*	1.00	**92.75**	**0.0451**	**94.91**	**87.86**	**91.52**	**0.1430**
	PostV	0.70	92.50	0.0528	94.89	87.79	91.46	0.1454
	PreV	0.70	92.41	0.0527	94.69	87.59	91.44	0.1491

the unpruned model by 0.16 points. Second, the evaluated generic pruning methods remain limited in document parsing, with all scoring substantially below the unpruned model in Overall score. These results suggest that generic token importance measures do not adequately reflect the importance of tokens within document text regions, limiting their effectiveness when directly applied to document parsing. Finally, T^3-Doc-PreV maintains competitive parsing accuracy while reducing the number of tokens processed by the visual encoder. Its Overall score of 92.02 is 0.63 points below baseline but exceeds PruMerge and PixelPrune by 0.44 and 3.04 points, respectively. It also achieves the lowest text edit distance among all evaluated methods, including the unpruned model. In contrast, Random (PreV) attains an Overall score of only 61.48, highlighting the importance of selecting informative tokens before visual encoding. Together, these results demonstrate the effectiveness of our text-aware token selection in largely preserving parsing accuracy while supporting token pruning both before and after the visual encoder.

Generalization across VLMs. To evaluate the generalizability of T^3-Doc, we apply it to representative document parsing models, including OvisOCR2 (Lu et al., 2026), WeVisDoc-4B (Yu et al., 2026), and Logics-Parsing-v2 (Logics-MLLM Team, 2026). Table 2 compares both pruning variants with their respective unpruned baselines at a token retention ratio of 70%. First, T^3-Doc largely preserves the original parsing performance across all three backbones while removing approximately 30% of visual tokens, demonstrating the applicability of text-aware pruning to different document VLMs. Second, the performance gap between the two pruning placements is relatively small, with T^3-Doc-PreV scoring only 0.09–0.26 Overall points below T^3-Doc-PostV. This indicates that pre-encoder pruning reduces the number of tokens processed by the visual encoder with only a modest additional decrease in parsing accuracy. Together, these results demonstrate the applicability of T^3-Doc across different backbones and show that flexible pruning placement offers a trade-off between parsing accuracy and visual encoding cost.

4.3 Computational Efficiency

Computational Cost Reduction. Table 3 compares the computational efficiency of the unpruned baseline, T^3-Doc-PostV, and T^3-Doc-PreV on a single NVIDIA H800 GPU, with maximum output lengths ranging from 512 to 4,096 tokens. Based on the results, we draw the following conclusions. First, T^3-Doc effectively reduces inference latency, with both variants achieving reductions of 28.8%–30.9% in time to first token (TTFT) across all evaluated settings. Second, T^3-Doc reduces KV cache usage, with both variants decreasing the prefill KV cache from 425.8 MB to 283.9 MB, a reduction of 33.3%. Finally, T^3-Doc-PreV further reduces computational cost. Across all evaluated settings, T^3-Doc-PreV achieves greater FLOPs reductions than T^3-Doc-PostV because it also reduces the number of tokens processed by the visual encoder. These results demonstrate that T^3-Doc reduces both inference latency and KV cache usage, with pre-encoder pruning providing additional computational savings.

Table 3: Efficiency of the two variants at a matched budget for different maximum output lengths. Detector latency is excluded.

Method	Output Length	TTFT (ms)		$KV\text{-}Cache_{pre}$ (MB)		$KV\text{-}Cache_{tot}$ (MB)		TFLOPs	
		Value	Δ	Value	Δ	Value	Δ	Value	Δ
baseline	512	51.8	-	425.8	-	481.7	-	28.06	-
T^3-Doc-PostV	512	36.9	-28.8%	283.9	-33.3%	339.8	-29.5%	22.32	-20.5%
T^3-Doc-PreV	512	**36.7**	-29.2%	**283.9**	-33.3%	**339.8**	-29.5%	**18.10**	-35.5%
baseline	1024	51.7	-	425.8	-	537.7	-	30.05	-
T^3-Doc-PostV	1024	36.5	-29.4%	283.9	-33.3%	395.8	-26.4%	24.16	-19.6%
T^3-Doc-PreV	1024	**36.4**	-29.6%	**283.9**	-33.3%	**395.8**	-26.4%	**19.94**	-33.6%
baseline	2048	52.1	-	425.8	-	649.7	-	34.21	-
T^3-Doc-PostV	2048	**36.0**	-30.9%	283.9	-33.3%	507.8	-21.8%	28.02	-18.1%
T^3-Doc-PreV	2048	36.4	-30.1%	**283.9**	-33.3%	**507.8**	-21.8%	**23.80**	-30.4%
baseline	4096	51.9	-	425.8	-	873.7	-	43.26	-
T^3-Doc-PostV	4096	**36.2**	-30.3%	283.9	-33.3%	731.8	-16.2%	36.45	-15.7%
T^3-Doc-PreV	4096	**36.2**	-30.3%	**283.9**	-33.3%	**731.8**	-16.2%	**32.23**	-25.5%

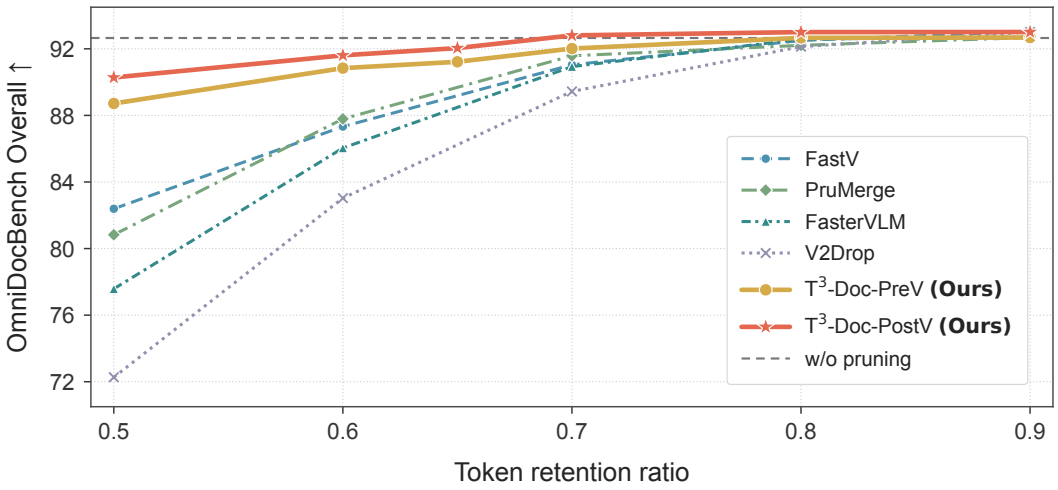


Figure 4: Performance under different token retention ratios. T^3-Doc consistently outperforms generic token-pruning baselines across retention budgets, with T^3-Doc-PostV remaining closest to the unpruned model and T^3-Doc-PreV providing competitive performance with earlier pruning.

Compatibility with Existing Acceleration Techniques. T^3-Doc constructs retention masks without relying on attention scores or modifying the VLM architecture, facilitating integration with existing acceleration techniques such as FlashAttention (Dao et al., 2022). Table 4 compares the parsing performance of T^3-Doc using PyTorch SDPA and FlashAttention. Switching to FlashAttention changes the Overall score by only 0.07 points, with similarly small differences in element-specific metrics. These results confirm the compatibility of T^3-Doc with FlashAttention.

4.4 Ablation Study

Effect of Token Retention Ratio. Figure 4 compares parsing performance at different token retention ratios. First, as the retention ratio decreases, both variants of T^3-Doc exhibit more gradual performance degradation than the compared generic pruning methods, and their advantages become more pronounced. This indicates that text-aware selection more effectively preserves information essential for parsing under limited token budgets. Second, T^3-Doc-PostV consistently achieves higher Overall scores than T^3-Doc-PreV in this experiment, suggesting that retaining full-image context during visual encoding benefits subsequent parsing. Nevertheless, both variants remain competitive at lower retention ratios, supporting the effectiveness of text-aware selection at either pruning stage.

Table 4: Comparison between FlashAttention and SDPA implementations.

Implementation	**Overall**↑	**Text**$^{\text{Edit}}$↓	**Formula**$^{\text{CDM}}$↑	**Table**$^{\text{TEDS}}$↑	**Table**$^{\text{TEDS-S}}$↑	**Reading Order**$^{\text{Edit}}$↓
SDPA	92.49	0.0446	94.84	87.08	90.25	0.1342
Flash-Attn	92.42	0.0452	94.82	86.94	90.10	0.1348
Δ	-0.07	$+0.0006$	-0.02	-0.14	-0.15	$+0.0006$

Table 5: Ablation study on the choice of text detector for T^3-Doc-PostV on OmniDocBench. All experiments use a target mean token retention ratio of 70%, with all settings held constant except the text detector. PP-OCRv6-Medium is the default detector. The best result in each column is highlighted in **bold**.

Detector	**Overall**↑	**Text**$^{\text{Edit}}$↓	**Formula**$^{\text{CDM}}$↑	**Table**$^{\text{TEDS}}$↑	**Table**$^{\text{TEDS-S}}$↑	**Reading Order**$^{\text{Edit}}$↓
Surya	92.42	0.0440	94.57	87.07	90.11	0.1356
PP-OCRv6-Tiny	91.74	0.0436	93.99	85.59	88.82	0.1367
PP-OCRv6-Small	92.42	0.0426	94.00	87.53	90.67	0.1350
PP-OCRv6-Medium	**92.81**	**0.0416**	**94.94**	**87.63**	**90.74**	**0.1317**

Finally, as the retention ratio increases, the performance gaps between methods narrow. T^3-Doc-PostV already achieves performance comparable to the unpruned baseline at the default retention ratio of 70%. At retention ratios of 80% and 90%, both variants perform slightly above the unpruned baseline. This observation suggests that removing irrelevant visual information can help reduce interference and improve parsing accuracy. However, retaining more tokens yields only limited gains over the default setting. These results support 70% as the default retention ratio to balance token reduction and parsing accuracy.

Effect of the Text Detector. To evaluate the effect of detector choice, we replace the default PP-OCRv6-Medium with its lighter Small and Tiny variants and Surya (Paruchuri & Datalab Team, 2025). The results are presented in Table 5. PP-OCRv6-Medium achieves the best performance, supporting its use as the default detector. PP-OCRv6-Small and Surya both achieve an Overall score of 92.42, 0.39 points below Medium, while PP-OCRv6-Tiny scores 91.74, a decrease of 1.07 points, indicating relatively modest overall differences. These results show that detector choice affects parsing performance, but T^3-Doc remains effective across different detectors. Detection errors can result in incomplete text-region coverage, while Coverage-Adaptive Thresholding expands coverage through region dilation and bounding-box merging, helping include some missed text regions and mitigate the impact of detection errors on parsing performance.

5 Conclusion

This work presents T^3-Doc, a text-aware, training-free token pruning framework for efficient document parsing. The results demonstrate that text-region-guided token selection better preserves parsing accuracy than selection based on generic token importance measures. On OmniDocBench, T^3-Doc removes approximately 30% of visual tokens on average, with T^3-Doc-PostV exceeding the unpruned model by 0.16 Overall points and outperforming all evaluated pruning baselines. Efficiency measurements excluding detector latency show that T^3-Doc-PostV reduces time to first token by up to 30.9%, while T^3-Doc-PreV reduces FLOPs by up to 35.5%. Experiments across additional document VLM backbones and evaluations with FlashAttention further support the framework's applicability. Together, these findings establish text-region guidance as an effective means of improving the accuracy–efficiency trade-off in document parsing without additional training. Future work will explore integrating traditional image processing techniques to improve text-region recall and reducing detection overhead to enhance robustness and end-to-end efficiency.

Limitations. Although region dilation and bounding-box merging in our Coverage-Adaptive Thresholding help compensate for incomplete text coverage, T^3-Doc can still be affected by detection errors. Specifically, visual tokens corresponding to text regions that remain uncovered after these adjustments are pruned, potentially affecting parsing accuracy.

Reproducibility Statement

The main text describes T^3-Doc and its experimental setup, including retention mask construction, pruning placements, threshold calibration, and evaluation protocols. The implementation code, experiment configurations, and instructions for reproducing the reported results will be publicly released.

LLM Usage Statement

GPT-6 was used to improve the language and clarity of the manuscript.

References

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

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report, 2025.

Junjie Chen, Xuyang Liu, Zichen Wen, Yiyu Wang, Siteng Huang, and Honggang Chen. Variation-aware vision token dropping for faster large vision-language models, 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, 2024.

Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, Yue Zhang, Yubo Zhang, Jing Zhang, Jun Zhang, Xing Wei, Yi Liu, Dianhai Yu, and Yanjun Ma. Boosting document parsing efficiency and performance with coarse-to-fine visual processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16655–16665, June 2026.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Zizhong Ding, Junxian Li, Kai Liu, Shaoqiu Zhang, Xiao Xiao, Linghe Kong, and Yulun Zhang. ET-Prune: Evidence-aware dynamic budgeting for visual token pruning in text-rich mllms, 2026.

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 *International Conference on Learning Representations*, 2021.

Yumeng Li, Guang Yang, Hao Liu, Bowen Wang, and Colin Zhang. dots.ocr: Multilingual document layout parsing in a single vision-language model, 2025.

Xudong Liu, Bicheng Wan, and Yulin Jin. LayoutLite: Token-level implicit layout analysis for efficient document ocr, 2026.

Logics-MLLM Team. Logics-Parsing-v2: Advanced document parsing large vision-language model. `https://huggingface.co/Logics-MLLM/Logics-Parsing-v2`, 2026.

Shiyin Lu, Yinglun Li, Yu Xia, Yuhui Chen, An-Yang Ji, Jun-Peng Jiang, Qing-Guo Chen, Jianshan Zhao, En Lin, Haijun Li, Cheng Qin, Zhao Xu, and Weihua Luo. OvisOCR2 technical report. *arXiv preprint arXiv:2607.13639*, 2026.

Junbo Niu, Zheng Liu, Zhuangcheng Gu, Bin Wang, Linke Ouyang, Zhiyuan Zhao, Tao Chu, Tianyao He, Fan Wu, Qintong Zhang, Zhenjiang Jin, Guang Liang, Rui Zhang, Wenzheng Zhang, Yuan Qu, Zhifei Ren, Yuefeng Sun, Zirui Tang, Boyu Niu, Yuanhong Zheng, Dongsheng Ma, Ziyang Miao, Hejun Dong, Siyi Qian, Junyuan Zhang, Fangdong Wang, Jingzhou Chen, Xiaomeng Zhao, Liqun Wei, Wei Li, Shasha Wang, RuiLiang Xu, Yuanyuan Cao, Lu Chen, Qianqian Wu, Huaiyu Gu, Lindong Lu, Dechen Lin, Shenguanlin, Xuanhe Zhou, Linfeng Zhang,

Yuhang Zang, Xiaoyi Dong, Jiaqi Wang, Bo Zhang, Lei Bai, Pei Chu, Weijia Li, Jiang Wu, Lijun Wu, Zhenxiang Li, Guangyu Wang, Zhongying Tu, Chao Xu, Kai Chen, Bowen Zhou, Dahua Lin, Wentao Zhang, and Conghui He. MinerU2.5: A decoupled vision-language model for efficient high-resolution document parsing. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 13–42. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-industry.3. URL https://aclanthology.org/2026.acl-industry.3/.

Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, Jin Shi, Fan Wu, Pei Chu, Minghao Liu, Zhenxiang Li, Chao Xu, Bo Zhang, Botian Shi, Zhongying Tu, and Conghui He. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24838–24848, June 2025.

Vikas Paruchuri and Datalab Team. Surya: A lightweight document OCR and analysis toolkit. https://github.com/datalab-to/surya, 2025. GitHub repository.

Jake Poznanski, Aman Rangapur, Jon Borchardt, Jason Dunkelberger, Regan Huff, Daniel Lin, Aman Rangapur, Christopher Wilhelm, Kyle Lo, and Luca Soldaini. olmOCR: Unlocking trillions of tokens in PDFs with vision language models, 2025.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22857–22867, 2025.

Zihan Tang, Leqi Shen, Hui Chen, Ao Wang, Ben Wan, Yan Feng, Ke Zhang, Sicheng Zhao, Tongxuan Liu, and Guiguang Ding. FastOCR: Dynamic visual fixation via KV cache pruning for efficient document parsing, 2026.

Hunyuan Vision Team, Pengyuan Lyu, Xingyu Wan, Gengluo Li, Shangpin Peng, Weinong Wang, Liang Wu, Huawen Shen, Yu Zhou, Canhui Tang, Qi Yang, Qiming Peng, Bin Luo, Hower Yang, Xinsong Zhang, Jinnian Zhang, Houwen Peng, Hongming Yang, Senhao Xie, Longsha Zhou, Ge Pei, Binghong Wu, Rui Yan, Kan Wu, Jieneng Yang, Bochao Wang, Kai Liu, Jianchen Zhu, Jie Jiang, Linus, Han Hu, and Chengquan Zhang. HunyuanOCR technical report, 2025.

Bin Wang, Fan Wu, Linke Ouyang, Zhuangcheng Gu, Rui Zhang, Renqiu Xia, Bo Zhang, and Conghui He. Image over text: Transforming formula recognition evaluation with character detection matching. *arXiv preprint arXiv:2409.03643*, 2024a.

Nan Wang, Zhiwei Jin, Chen Chen, and Haonan Lu. PixelPrune: Pixel-level adaptive visual token reduction via predictive coding, 2026.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution, 2024b.

Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, Chunrui Han, and Xiangyu Zhang. General OCR theory: Towards OCR-2.0 via a unified end-to-end model, 2024.

Haoran Wei, Yaofeng Sun, and Yukun Li. DeepSeek-OCR: Contexts optical compression, 2025.

Hao Wu, Haoran Lou, Xinyue Li, Zuodong Zhong, Zhaojun Sun, Phellon Chen, Xuanhe Zhou, Kai Zuo, Yibo Chen, Xu Tang, Yao Hu, Boxiang Zhou, Jian Wu, Yongji Wu, Wenxin Yu, Yingmiao Liu, Yuhao Huang, Manjie Xu, Gang Liu, Yidong Ma, Zhichao Sun, and Changhao Qiao. FireRed-OCR technical report, 2026.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Conical visual concentration for efficient large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14593–14603, June 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802, June 2025.

Hao Yu, Kang Liu, Linnan Zhao, Jiabo Zhan, Chong Sun, Chen Li, and Jing Lyu. WeVisDoc: From coverage to capability for robust end-to-end document parsing. *arXiv preprint arXiv:2609.20423*, 2026.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20857–20867, October 2025a.

Rongyang Zhang, Chengqiang Lu, Cong Li, Hongchao Gu, Tingjia Shen, Xuyang Zhi, Qimeng Wang, Yan Gao, Yi Wu, Yao Hu, Hao Wang, and Enhong Chen. Multi-image visual token pruning in large visual language models, 2026a.

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74840–74857. PMLR, 2025b. URL https://proceedings.mlr.press/v267/zhang25s.html.

Yubo Zhang, Xueqing Wang, Manhui Lin, Yue Zhang, Penglongyi Deng, Ting Sun, Tingquan Gao, Zelun Zhang, et al. PP-OCRv6: From 1.5m to 34.5m parameters, surpassing billion-scale vlms on ocr tasks, 2026b.

Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. Image-based table recognition: data, model, and evaluation. In *European conference on computer vision*, pp. 564–580. Springer, 2020.