Task-Agnostic and Task-Conditioned Token Reduction in Vision Transformers: Accuracy–Efficiency Tradeoffs Under Fixed and Adaptive Retention

Chase Lambuth
Johns Hopkins University

Abstract

This research investigates how task-agnostic and task-conditioned token reduction strategies compare for vision transformer semantic segmentation and whether adaptive token retention can improve the accuracy–efficiency tradeoff over fixed token retention. Token reduction is important because it can affect training time, inference speed, hardware requirements, and deployment costs. In this study, the models are first trained on the full semantic segmentation task, and the semantic predictions are then converted into binary target masks so Macro Target IoU can be used to evaluate class-conditioned performance. The experiments compare task-agnostic and task-conditioned token reduction methods using the same lightweight PVTv2-style backbone, LoveDA dataset splits, and training setup. The task-agnostic baselines include full dense token processing, random Stage-2 pruning, learned visual-importance pruning, and a ToMe-style token merging proxy. Random Stage-2 pruning achieves performance comparable to the full-token model, with a Macro Target IoU score of 0.312 compared with 0.310 for the full-token model while retaining only 75% of Stage-2 tokens. The task-conditioned experiments compare discrete task IDs, learned task embeddings, and Natural-Language Routing. Under fixed 75% retention, Natural-Language Routing performs best among the task-conditioned methods with a Macro Target IoU of 0.286. Adaptive token retention improves both learned task embeddings and Natural-Language Routing while keeping average retention close to the same computational budget. Adaptive task embeddings had the strongest task-conditioned Macro Target IoU at 0.307, while adaptive Natural-Language Routing had a Macro Target IoU of 0.302 and the strongest semantic mIoU and Boundary IoU score among the task-conditioned models. However, neither adaptive method had a higher observed Macro Target IoU than the 0.312 achieved by random pruning. Because the current implementation uses dense masking instead of true sparse execution, the reported efficiency improvements represent analytical computational savings, not measured inference-time speedups. The results overall demonstrate that simple random pruning remains a strong baseline, while adaptive retention improves the effectiveness of task-conditioning strategies and reduces the performance gap between task-conditioned and task-agnostic token reduction.

Keywords: Vision Transformers, Semantic Segmentation, Task-Agnostic Token Reduction, Task-Conditioned Token Routing, Token Pruning, Vision-Language Models, Computational Efficiency

1. INTRODUCTION

Vision transformers have become one of the leading architectures for computer vision because their self-attention mechanism allows the modeling of long-range relationships between image regions. This ability has resulted in sizable improvements across computer vision tasks, including image classification, object detection, and semantic segmentation. However, it also introduces a high computational cost. Self-attention has quadratic computational complexity with respect to the number of tokens, which causes computation to increase rapidly as image resolution and token count grow [1], [2]. As transformer models continue to be applied to dense prediction tasks and resource-constrained settings, improving their computational efficiency has become an important area of research.

Numerous architectures have been proposed to improve the computational efficiency of vision transformers. The Pyramid Vision Transformer introduces hierarchical feature representations inspired by the multi-scale structure commonly used in convolutional neural networks and uses spatial-reduction attention to reduce the computational cost of self-attention [1]. These approaches demonstrate that the computational cost of vision transformers can be reduced through more efficient architectural designs while maintaining strong performance on dense prediction tasks.

There has also been substantial research on token reduction techniques for improving the computational efficiency of vision transformers. These methods reduce computation by decreasing the number of visual

tokens that must be processed in later transformer layers. DynamicViT dynamically prunes less informative tokens based on their estimated importance [3], while EViT preserves attentive tokens and reorganizes less informative tokens through fusion [4]. Token Merging (ToMe) similarly reduces token counts by progressively merging similar tokens [5]. More recent work has also investigated frequency-aware token reduction to preserve important information while reducing computational overhead [2]. Together, these studies demonstrate that substantial redundancy can exist among visual tokens and that token reduction can provide favorable tradeoffs between computational efficiency and predictive performance.

Many token reduction strategies are task-agnostic because token importance is determined from visual information without explicitly considering the specific task being performed. Semantic segmentation provides a particularly interesting setting for comparing these methods with task-conditioned token reduction because the relevance of an image region can depend on the requested segmentation task. For example, when identifying roads, many image regions containing buildings or vegetation may contribute relatively little to the requested prediction, while those same regions may become important when segmenting buildings or forests. Recent work on task-oriented segmentation supports this idea. VLTP uses vision-language guidance to estimate the task-specific relevance of image tokens and selectively passes highly relevant tokens to deeper transformer layers, demonstrating that token pruning can be conditioned on the requested segmentation task [6]. This creates an important comparison between task-agnostic strategies that reduce tokens based on visual information and task-conditioned strategies that also incorporate information about the requested task.

Task information can be represented in several ways. One approach is a discrete task ID, which provides the model with a numerical identifier specifying the requested segmentation task. Another approach is a learned task embedding, where the model learns a trainable vector representation for each task during training. A third approach is Natural-Language Routing, where the requested task is represented using language instead of only using a numerical identifier. Vision-language models such as CLIP have demonstrated that natural-language supervision can produce representations that connect textual concepts with visual information [7], while VLTP demonstrates that vision-language guidance can be directly incorporated into task-oriented token pruning for semantic segmentation [6]. Comparing these representations with task-agnostic token reduction provides a way to determine whether incorporating task information produces a better accuracy–efficiency tradeoff and, if so, which type of task representation is most effective.

To address this problem, this research compares task-agnostic and task-conditioned token reduction strategies under a consistent experimental framework for semantic segmentation. The task-agnostic comparison includes random token pruning, visual-importance pruning, and ToMe-style token merging, while the task-conditioned comparison evaluates discrete task IDs, learned task embeddings, and Natural-Language Routing. The fixed-retention experiments use the same nominal token-retention ratio so that the different strategies can be compared under a consistent computational budget. In addition, adaptive token retention is evaluated to determine whether allowing the retained-token budget to change across image–task pairs can improve the accuracy–efficiency tradeoff compared with applying the same fixed retention ratio to every input. Together, these experiments evaluate how task-agnostic and task-conditioned token reduction compare and whether adaptive retention provides an additional advantage over fixed token reduction.

2. RELATED WORK

Vision transformers have demonstrated strong performance across a wide range of computer vision applications by replacing convolutional operations with self-attention. Variants such as Pyramid Vision Transformers and Swin Transformer have successfully reduced computational cost without sacrificing performance [1], [8]. Despite these advances, transformer-based models remain computationally demanding because self-attention is still applied across large numbers of visual tokens.

A method that has been developed to reduce computational cost for vision transformers is token reduction. Token reduction techniques generally fall into two main categories: token pruning, which removes the least informative tokens, and token merging, which combines similar tokens into a single representation [2]. In 2021, DynamicViT [3] demonstrated that only a subset of image tokens contribute meaningfully to the performance of vision transformers and proposed pruning tokens that contribute little to the final prediction. This reduces computational cost because the computational complexity of self-attention is directly related to the number of tokens, scaling as $O(N^2D)$, where N is the number of tokens and D is the embedding dimension.

Following DynamicViT, several token reduction methods were proposed. EViT [4] introduced tech-

niques that dynamically reorganize or remove tokens during inference, while token merging methods such as ToMe [5] combine visually similar tokens to reduce the computational complexity of later transformer layers. These approaches have shown that significant reductions in computation can be achieved while maintaining strong performance, however, token importance is generally determined from visual features alone, meaning that all segmentation tasks receive the same computational allocation regardless of the object class being predicted.

More recently, research has investigated whether task information can directly guide token reduction. Vision-language-guided token pruning (VLTP) [6] proposed using natural-language descriptions to identify task-relevant visual tokens for semantic segmentation. The model uses language guidance to prioritize tokens that are most relevant to the segmentation task while pruning less informative regions. Unlike earlier token reduction methods, VLTP does not rely solely on visual features. This demonstrated that semantic task information can be incorporated into the token-pruning process to improve computational efficiency while maintaining high segmentation performance. However, VLTP focuses exclusively on natural-language guidance and does not directly compare task-conditioned routing with multiple task-agnostic token reduction strategies under the same experimental framework. It also does not investigate whether other forms of task conditioning, such as discrete task identifiers or learned task embeddings, provide different tradeoffs between segmentation accuracy and efficiency, or whether the amount of retained computation should adapt to each image and segmentation task.

Vision-language models have demonstrated that natural-language information can guide visual representations and improve transfer across tasks. Models such as CLIP [7] learn shared visual and textual representations that enable semantic relationships captured in language to influence visual understanding. These advances suggest that task information can provide valuable guidance when allocating computational resources. However, there is limited work directly comparing task-agnostic and task-conditioned token reduction for semantic segmentation under the same routing framework and computational budget. This work addresses that gap by comparing task-agnostic random pruning, visual-importance pruning, and token merging with task-conditioned routing using discrete task IDs, learned task embeddings, and natural-language descriptions. These methods are evaluated under consistent model architectures, training procedures, and computational budgets. As an additional extension, this work investigates adaptive token retention to determine whether allowing the retained-token budget to change across image–task pairs can improve the accuracy–efficiency tradeoff compared with fixed token retention.

3. RESEARCH PROBLEM

Token reduction methods can reduce the computational requirements of vision transformers by removing, merging, or reorganizing visual tokens that contribute less to the final prediction [3], [4], [5]. Many of these methods determine token importance using visual information without considering the specific task being performed. This task-agnostic approach can be effective because visual redundancy allows computation to be reduced without requiring additional information about the requested prediction. However, in semantic segmentation, the relevance of individual image regions can also change depending on the object class being segmented. Recent work on vision-language-guided token pruning has similarly identified task relevance as an important consideration for token selection in task-oriented segmentation [6]. This creates an important question of whether token reduction is more effective when it remains task-agnostic or when the routing decision is explicitly conditioned on the requested segmentation task.

This comparison is important because token reduction directly affects the amount of computation required by transformer models. Computational cost affects training time, inference speed, hardware requirements, and deployment costs, particularly as image resolution and model size increase. Previous token reduction methods have demonstrated that redundant visual tokens can be removed or merged while preserving much of the original model performance [3], [5]. These reductions can be especially important for resource-constrained and edge-computing applications, where available processing power, memory, and energy may be limited. In these settings, reducing unnecessary token computation can make transformer models more practical while also reducing the latency associated with processing large collections of visual tokens.

Reducing token computation is also important during model development and large-scale deployment. Training transformer models can require advanced computational resources, particularly for high-resolution vision tasks and large datasets. At deployment time, these requirements can increase operational costs and energy consumption when inference must be performed repeatedly or across many devices. Token reduction therefore provides a way to investigate whether transformer computation

can be allocated more efficiently while maintaining the spatial information required for dense prediction.

This research compares task-agnostic and task-conditioned token reduction strategies under a consistent semantic segmentation framework. The task-agnostic methods include random Stage-2 pruning, learned visual-importance pruning, and ToMe-style token merging, while the task-conditioned methods use discrete task IDs, learned task embeddings, and natural-language task representations to guide token selection. All fixed-retention methods are evaluated using the same Stage-2 token budget, allowing differences in segmentation performance to be compared without changing the nominal amount of token reduction. The central research question is how task-agnostic and task-conditioned token reduction strategies compare for semantic segmentation, and whether adaptive token retention can further improve the accuracy–efficiency tradeoff over fixed token retention.

The experimental study is designed to determine whether incorporating task information provides an advantage over task-agnostic token reduction and whether this depends on the representation used to describe the task. Natural-language and learned task representations are expected to provide richer information for task-conditioned routing than discrete task IDs, while task-agnostic methods provide an important comparison for determining whether this additional conditioning is necessary. Adaptive token retention is also evaluated to determine whether allowing the retained-token budget to change across image–task pairs can improve the accuracy–efficiency tradeoff compared with applying the same fixed retention ratio to every input.

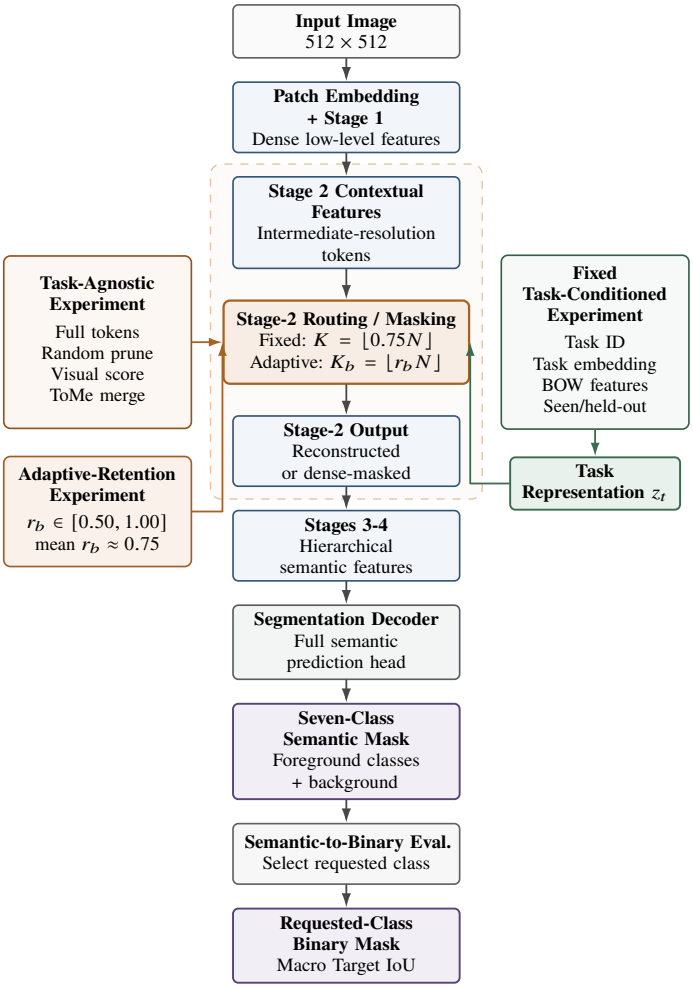


Figure 1. Current semantic-to-binary experimental protocol. The experiments train each transformer variant as a full semantic segmentation model rather than as a direct binary-mask predictor. The task-agnostic experiment compares Stage-2 token policies: full-token processing, random pruning, visual-importance pruning, and ToMe-style merging. The fixed task-conditioned experiment keeps the same semantic-first objective and fixed nominal 75% Stage-2 retention while adding task-conditioned dense masking from a discrete task ID, learned task embedding, or bag-of-words prompt features. The adaptive-retention experiment extends the same router with adaptive retention, predicting a sample-specific retention ratio $r_b \in [0.50, 1.00]$ and retaining $K_b = \lfloor r_b N \rfloor$ Stage-2 tokens while regularizing the mean retention toward 0.75. During evaluation, the predicted semantic mask is converted into a requested-class binary mask, and Macro Target IoU is computed over the six foreground LoveDA classes.

4. METHODS

The experimental framework evaluates token reduction within a semantic-first segmentation pipeline. Each model predicts a seven-class semantic segmentation mask, after which the semantic prediction is converted into a binary mask for the requested foreground class. This design keeps the underlying segmentation objective consistent across all experiments while allowing class-conditioned performance to be measured using Macro Target IoU.

Token intervention is applied at Stage 2 of a lightweight PVTv2-style hierarchical transformer backbone. Three experimental groups are considered. First, task-agnostic baselines compare full-token processing with random pruning, visual-importance pruning, and ToMe-style token merging. Second, fixed task-conditioned routing incorporates a discrete task identifier, learned task embedding, or Natural-Language Routing using bag-of-words prompt features while retaining a nominal 75% of Stage-2 tokens. Third, adaptive routing predicts a sample-specific retention ratio while regularizing the average token budget toward the same nominal 75% target.

Figure 1 summarizes the complete experimental pipeline.

4.1. Semantic-First Segmentation Framework

All experimental configurations are trained using the same semantic segmentation objective. For an input image x, the segmentation model produces a dense seven-class prediction

$$\hat{Y} = f_\theta(x), \tag{1}$$

where the output classes correspond to the six foreground LoveDA categories and background. Training the models on the complete semantic task prevents the routing mechanism from changing the fundamental output space between requested classes and provides a common representation for all experimental conditions.

During class-conditioned evaluation, a requested foreground class t is selected and the semantic prediction is converted to a binary mask according to

$$\hat{M}_t(p) = \mathbb{I}\left[\arg\max_c \hat{Y}_{p,c} = t\right], \tag{2}$$

where p denotes a pixel location and $\mathbb{I}[\cdot]$ is the indicator function. The corresponding ground-truth semantic mask is converted using the same requested class. This semantic-to-binary projection allows class-conditioned performance to be measured without training a separate binary segmentation model for each task.

4.2. Backbone and Stage-2 Intervention

The experiments use the same lightweight PVTv2-style hierarchical vision transformer backbone across all configurations. Input images are resized to 512×512 pixels and processed through an initial patch embedding stage followed by four hierarchical transformer stages.

Token reduction is applied at Stage 2. This location provides a compromise between spatial resolution and semantic abstraction: Stage-2 features contain more contextual information than the earliest features while still containing substantially more tokens than the deeper transformer stages. The routing experiments therefore modify the Stage-2 token representation while leaving the remainder of the segmentation architecture unchanged.

Let

$$X^{(2)} = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{N \times D} \tag{3}$$

denote the Stage-2 token sequence, where N is the number of tokens and D is the embedding dimension. A routing policy assigns each token a selection decision or importance score before the representation is passed to the remaining transformer stages.

4.3. Task-Agnostic Token Reduction

The first experimental group establishes task-agnostic baselines. These methods do not receive information about the requested segmentation class and therefore determine token reduction entirely from the visual representation or from a class-independent sampling rule.

The full-token configuration processes all Stage-2 tokens and serves as the dense reference model. Random pruning retains a uniformly sampled subset of tokens and provides a simple control for determining whether more structured token-selection strategies provide benefits beyond reducing the token count itself.

Visual-importance pruning assigns each Stage-2 token an importance score derived from the visual features and retains the highest-scoring subset. The ToMe-style configuration instead approximates token merging by combining visually similar representations before later transformer processing.

For all fixed reduced-token configurations, the nominal retention ratio is

$$r = 0.75, \tag{4}$$

giving

$$K = \lfloor 0.75N \rfloor \tag{5}$$

retained Stage-2 tokens.

4.4. Task-Conditioned Routing

The second experimental group evaluates whether knowledge of the requested segmentation class can improve token selection. Each image–task pair is associated with a task representation z_t, which is combined with the Stage-2 visual representation to produce task-conditioned token scores.

Three task representations are evaluated. The discrete task-ID configuration supplies the router with an integer identifier for the requested foreground class. The learned-embedding configuration maps each task identifier to a trainable vector representation. The Natural-Language Routing conditions the router on lightweight bag-of-words prompt features constructed from textual descriptions of the segmentation task.

Conceptually, the router computes

$$s_i = g_\phi(x_i, z_t), \tag{6}$$

where s_i is the routing score for Stage-2 token x_i, z_t represents the requested task, and g_ϕ denotes the learned routing function.

Under fixed retention, the tokens with the highest routing scores are selected subject to

$$K = \lfloor 0.75N \rfloor. \tag{7}$$

The segmentation objective remains seven-class semantic segmentation; task information affects the routing mechanism rather than changing the decoder output into a binary predictor.

4.5. Adaptive Token Retention

The third experimental group removes the requirement that every image–task pair retain exactly the same

fraction of Stage-2 tokens. Instead, the adaptive router predicts a retention ratio r_b for each sample b,

$$r_b \in [0.50, 1.00]. \tag{8}$$

The number of retained tokens becomes

$$K_b = \lfloor r_b N \rfloor . \tag{9}$$

This allows the model to allocate additional computation to image–task pairs that require more visual information while reducing the token budget for easier cases.

To keep the adaptive models comparable with the fixed 75% retention experiments, training includes a budget regularization term encouraging the mean predicted retention ratio to remain close to 0.75. A generic form of this penalty is

$$\mathcal{L}_{\text{budget}} = \left(\frac{1}{B} \sum_{b=1}^{B} r_b - 0.75 \right)^2 , \tag{10}$$

where B is the batch size.

The total optimization objective can therefore be expressed as

$$\mathcal{L} = \mathcal{L}_{\text{seg}} + \lambda_{\text{budget}} \mathcal{L}_{\text{budget}}, \tag{11}$$

where $\mathcal{L}_{\text{seg}}$ denotes the semantic segmentation loss and λ_{budget} controls the strength of the retention-budget regularization.

4.6. Experimental Setup and Training Details

The LoveDA dataset contained 2,522 training images, 1,669 validation images, and 1,796 test images in the local dataset audit. Because the official LoveDA test split does not provide labels in the project directory, the experiments trained on the labeled training split and partitioned the labeled LoveDA validation split into non-overlapping validation and test subsets. This produced 667 validation images and 1,001 test images. For class-conditioned evaluation, each validation and test image was expanded across the six foreground target classes, while training used one deterministic image–task pairing per image per epoch.

All models used a random seed of 42, input images resized to 512×512, a batch size of 4, and 10 training epochs with early stopping patience of 4 epochs. Training used AdamW with a learning rate of 1×10^{-4} and weight decay of 1×10^{-4}, followed by a polynomial learning-rate decay schedule. The segmentation loss was cross-entropy with the ignore index excluded from the loss. For adaptive retention, $\lambda_{\text{budget}} = 0.02$. Training images used resize, random horizontal and vertical flips, random 90° rotations, shift-scale-rotate augmentation, brightness/contrast jitter, Hue-Saturation-Value jitter, ImageNet normalization, and tensor conversion. Validation and test images used only resizing, normalization, and tensor conversion.

Each run saved a latest checkpoint after every epoch and also saved a separate best checkpoint whenever validation Macro Target IoU improved. Final model selection used the best checkpoint when available; otherwise the latest checkpoint was used. Checkpoints stored the model, optimizer, scheduler, scaler, training history, best validation score, epoch number, data split fractions, batch size, epoch count, and early stopping state.

4.7. Natural-Language Prompt Transfer

Natural-Language Routing is additionally evaluated under a prompt-transfer setting. During training, each LoveDA foreground class is associated with a set of task descriptions. At evaluation time, these descriptions are replaced with held-out paraphrases that refer to the same semantic classes.

The image data, semantic labels, model parameters, and evaluation procedure remain unchanged. Consequently, this experiment measures the sensitivity of the lightweight Natural-Language Routing model to alternative descriptions of known tasks rather than open-vocabulary transfer to previously unseen semantic categories.

4.8. Evaluation Metrics

The primary evaluation metric is Macro Target IoU. For each requested foreground class t, the semantic model prediction is converted into the corresponding binary target mask. Intersection over Union is then computed as

$$\text{IoU}_t = \frac{\left| \hat{M}_t \cap M_t \right|}{\left| \hat{M}_t \cup M_t \right|}. \tag{12}$$

Macro Target IoU averages the requested-class IoU across the six foreground LoveDA classes so that each task contributes equally to the final score.

Additional metrics include task-present Macro Target IoU, semantic mIoU, foreground mIoU, Boundary IoU, and pixel accuracy. Task-present IoU includes only image–task pairs for which the requested foreground class is present in the ground-truth image. Semantic mIoU evaluates the underlying seven-class segmentation output, while foreground mIoU excludes the background category. Boundary IoU measures segmentation quality near class boundaries, and pixel accuracy reports the fraction of correctly classified valid pixels.

4.9. Analytical Efficiency

The computational comparison is based on the number of tokens retained from the Stage 2 representation.

Because self-attention has approximately quadratic complexity with respect to token count, retaining a fraction r of the routed token representation gives an idealized relative attention-token ratio of

$$C_{\text{ideal}} \propto r^2. \tag{13}$$

For the fixed retention ratio $r = 0.75$, this corresponds to

$$0.75^2 = 0.5625, \tag{14}$$

or approximately 56.25% of the full-token attention cost for any subsequent sparse attention operating only on the retained tokens.

The current implementation uses dense masking or reconstruction rather than hardware-level sparse execution. Therefore, the reported efficiency values represent analytical token-retention and idealized attention-scaling estimates, not direct measurements of full Stage-2 attention savings, wall-clock latency, memory use, or inference time.

5. RESULTS

The proposed framework was evaluated through three groups of experiments. First, the full-token model and task-agnostic token-reduction methods were compared to determine how Stage-2 token reduction affects segmentation performance without task information. Second, the three task-conditioned routing strategies were evaluated under the same fixed 75% Stage-2 token budget. Finally, adaptive token retention and prompt transfer were evaluated to measure performance when the retained-token count varies by input and when natural-language task descriptions differ from those used during training.

5.1. Fixed Token Reduction

The fixed-retention experiments compare seven configurations under the same semantic-first evaluation protocol. The full-token model provides the dense transformer reference, while random pruning, visual-importance pruning, and ToMe-style merging provide task-agnostic token-reduction baselines. These models are compared with task-conditioned routing using discrete task IDs, learned task embeddings, and Natural-Language Routing. All reduced-token configurations use a nominal Stage-2 retention ratio of 0.75.

Table 1 reports the primary segmentation metrics for these configurations. Macro Target IoU measures requested-class overlap after semantic-to-binary conversion and serves as the primary comparison metric. Present IoU reports Macro Target IoU only for image–task pairs in which the requested class is present, while semantic mIoU measures performance on the underlying seven-class semantic segmentation task.

Table 1. Primary held-out test performance for the full-token baseline, task-agnostic token-reduction methods, and fixed task-conditioned routing strategies. Target IoU reports Macro Target IoU across requested classes, Present IoU includes only image–task pairs where the requested class is present, and semantic mIoU measures seven-class segmentation performance.

Method	Target IoU	Present IoU	Sem. mIoU
Full Token	0.310	0.328	0.330
Random Pruning	**0.312**	**0.330**	**0.331**
Visual Pruning	0.308	0.325	0.328
ToMe-Style	0.298	0.314	0.319
Task ID	0.269	0.287	0.282
Task Embedding	0.276	0.296	0.300
Natural-Language Routing	0.286	0.305	0.308

Random pruning achieved the highest Macro Target IoU among the fixed configurations at 0.312, closely matching the full-token model at 0.310. Among the task-conditioned methods, Natural-Language Routing performed best at 0.286.

Table 2 reports the additional segmentation metrics, including foreground mIoU, Boundary IoU, and pixel accuracy.

Table 2. Additional held-out test segmentation metrics for the full-token baseline, task-agnostic token-reduction methods, and fixed task-conditioned routing strategies. Foreground mIoU is computed across the six foreground LoveDA classes, Boundary IoU measures segmentation quality around class boundaries, and pixel accuracy reports the proportion of correctly classified valid pixels.

Method	FG mIoU	Boundary IoU	Pixel Acc.
Full Token	0.310	0.114	0.562
Random Pruning	**0.312**	**0.115**	**0.564**
Visual Pruning	0.308	0.113	0.556
ToMe-Style	0.298	0.112	0.558
Task ID	0.269	0.101	0.500
Task Embedding	0.276	0.104	0.542
Natural-Language Routing	0.286	0.108	0.540

Random pruning produced the strongest additional metrics among the reduced-token baselines, closely matching the full-token model. Among the task-conditioned models, Natural-Language Routing achieved the highest foreground mIoU and Boundary IoU, while learned task embeddings achieved the highest pixel accuracy.

All fixed reduced-token configurations use a nominal Stage-2 retention ratio of 0.75, corresponding to the analytical attention-cost estimate described in

Section 4.9. Because the current implementations use dense reconstruction or masking, these values represent analytical token-count reductions rather than measured sparse-execution speedups.

5.2. Adaptive Token Retention

The adaptive-retention experiment extends each task-conditioned routing strategy by allowing the retained Stage-2 token count to vary across image–task pairs. The adaptive router predicts a retention ratio between 0.50 and 1.00, while a regularization term encourages average retention to remain near the fixed 0.75 target. Each adaptive configuration is therefore compared with the corresponding fixed-retention model using the same conditioning representation and evaluation procedure.

Table 3 reports the primary class-conditioned metrics for the fixed and adaptive variants. The Fixed columns contain the results obtained using a constant 75% Stage-2 retention ratio, while the Adaptive columns contain the results obtained when the retained-token ratio is predicted separately for each image–task pair. Both Macro Target IoU and Present IoU are reported to compare performance across all requested tasks and across only those cases in which the target class is present.

Table 3. Comparison of fixed and adaptive task-conditioned routing on the held-out test set. Fixed models use a constant 75% Stage-2 token-retention ratio, while adaptive models predict the retained-token ratio separately for each image–task pair. Results are reported using Macro Target IoU and task-present Macro Target IoU.

Conditioning	Target IoU		Present IoU	
	Fixed	Adapt.	Fixed	Adapt.
Task ID	0.269	0.268	0.287	0.287
Embedding	0.276	**0.307**	0.296	**0.324**
Natural-Language Routing	0.286	0.302	0.305	0.316

Adaptive retention produced the largest improvement for learned task embeddings, increasing Macro Target IoU from 0.276 to 0.307. Natural-Language Routing also improved from 0.286 to 0.302, while task-ID routing remained approximately unchanged at 0.269 and 0.268 under fixed and adaptive retention, respectively.

Table 4 reports the additional segmentation metrics for the adaptive configurations together with their observed retention ratios. Semantic mIoU measures seven-class segmentation performance, Boundary IoU evaluates segmentation overlap near class boundaries, and pixel accuracy measures the proportion of correctly classified valid pixels. Unlike the nominal fixed-retention ratio, the retention values reported in this table are calculated from the actual token counts selected by each adaptive router over the held-out test set.

Table 4. Additional held-out test results for adaptive task-conditioned routing. Semantic mIoU, Boundary IoU, and pixel accuracy report segmentation performance, while Ret. gives the observed mean Stage-2 token-retention ratio selected by each adaptive router across the test set.

Conditioning	Sem. mIoU	Boundary IoU	Pixel Acc.	Ret.
Task ID	0.287	0.103	0.518	0.759
Embedding	0.322	0.109	0.544	0.765
Natural-Language Routing	**0.325**	**0.115**	**0.565**	0.765

Adaptive Natural-Language Routing achieved the highest semantic mIoU, Boundary IoU, and pixel accuracy among the adaptive configurations, followed by learned task embeddings and task-ID routing.

Observed average retention remained close to the nominal 0.75 target for all three adaptive configurations. Task-ID routing retained an average ratio of 0.759, while learned task embeddings and Natural-Language Routing each retained approximately 0.765 of the Stage-2 tokens.

5.3. Prompt-Transfer Generalization

The final experiment evaluates the Natural-Language Routing models using alternative descriptions of the known LoveDA classes. During training, each class is associated with a set of seen task descriptions. Prompt-transfer evaluation replaces these descriptions with held-out paraphrases while keeping the image data, semantic classes, model parameters, and evaluation procedure unchanged. Both the fixed and adaptive Natural-Language Routing models are evaluated under the seen and held-out prompt conditions.

Table 5 compares Macro Target IoU under the seen and held-out prompt conditions. This evaluation is performed separately for fixed and adaptive Natural-Language Routing to determine whether changing the task description affects class-conditioned segmentation performance.

Table 5. Prompt-transfer performance for fixed and adaptive Natural-Language Routing on the held-out test set. Seen prompts use task descriptions represented during training, while unseen prompts use held-out alternative descriptions of the same LoveDA classes.

Routing	Seen Target IoU	Unseen Target IoU
Fixed	0.2864	0.2861
Adaptive	0.3020	0.3020

Macro Target IoU remained nearly unchanged between the seen and held-out prompt conditions for both Natural-Language Routing configurations. Fixed routing changed from 0.2864 to 0.2861, while adaptive routing remained about 0.3020 under both prompt conditions.

6. DISCUSSION

6.1. Task-Agnostic Token Reduction Performance

The task-agnostic reduction experiments evaluate whether Stage-2 token reduction can maintain segmentation performance without using class-specific routing information. The full-token PVTv2-style model achieved a Macro Target IoU score of 0.310, while random token pruning had a similar score of 0.312 despite retaining only 75% of the Stage-2 tokens. Visual-importance pruning also remained close to the full-token baseline with a Macro Target IoU of 0.308, while the ToMe-style token merging proxy had the weakest task-agnostic result at 0.298.

These results suggest that moderate Stage-2 token reduction does not necessarily reduce segmentation performance in the current semantic-first setup. In particular, random pruning maintained performance similar to the full-token model while removing 25% of the Stage-2 tokens. Additionally, the result suggests that a portion of the Stage-2 visual tokens can be removed while only slightly affecting the final segmentation predictions. This supports the general motivation for token reduction, since processing every available token was not necessary to maintain the performance of the full-token model in these experiments.

Random pruning also provides an important comparison for evaluating the more structured token reduction methods since it had the highest observed Macro Target IoU score among the task-agnostic configurations. However, its performance remained close to both the full-token and visual-importance models. This means that simply reducing the number of tokens was already a strong strategy. These results do not demonstrate that more structured token selection is automatically better than generic token removal.

6.2. Task-Agnostic and Task-Conditioned Comparison

The task-conditioned experiments evaluate whether information about the requested segmentation class can provide a useful signal for deciding which visual tokens should be retained. Under fixed 75% retention, Natural-Language Routing produced the strongest performance of the three task-conditioned strategies, achieving a Macro Target IoU of 0.286. Learned task embeddings followed at 0.276, while discrete task IDs achieved 0.269.

The ordering of these results suggests that the representation used to describe the task affects the quality of the routing decisions. Natural-Language Routing provides more information than a discrete numerical task identifier, while learned task embeddings allow the model to develop a representation of each task during training. In the fixed experiment, Natural-Language Routing provided the strongest task-conditioned routing signal, although the difference between the three approaches was not large enough to outperform the task-agnostic methods.

The direct comparison between task-agnostic and task-conditioned reduction is important because the motivation for task-conditioned routing was that token relevance may depend on the requested segmentation task. While the results show differences between the three conditioning strategies, they do not show that task information alone is sufficient to improve token reduction. The strongest fixed task-conditioned model achieved a Macro Target IoU of 0.286 compared with 0.312 for random pruning and 0.308 for visual-importance pruning. Therefore, under fixed retention, the results favor the strongest task-agnostic strategies rather than showing an overall advantage for task-conditioned routing.

One possible explanation is that the routing problem is more difficult than simply identifying which regions appear relevant to the requested class. Because the underlying model is trained for full semantic segmentation, information that does not appear directly related to the requested class may still contribute useful context to the final prediction. Removing tokens based on the task representation could therefore remove contextual information that remains useful to the semantic segmentation model. Random pruning may preserve this information more evenly because it does not make a strong assumption about which regions are relevant to the requested class.

The Natural-Language Routing model also remained stable during prompt-transfer evaluation. Changing from the task descriptions used during training to held-out descriptions of the same classes produced almost no change in Macro Target IoU. This shows that the lightweight bag-of-words prompt features were not highly sensitive to the specific wording used in the prompt-transfer experiment. However, these prompts still describe known LoveDA classes, so this result should be interpreted as robustness to alternative descriptions rather than generalization to completely unseen segmentation tasks.

6.3. Fixed and Adaptive Token Retention

Adaptive token retention provides the second major comparison by evaluating whether the same token budget should be applied to every input. Instead of requiring every image–task pair to retain exactly 75% of the Stage-2 tokens, the adaptive models were allowed to determine their retention ratio based on the input while being regularized toward the same average computational budget.

The largest improvement occurred with learned task

embeddings. Macro Target IoU increased from 0.276 under fixed retention to 0.307 under adaptive retention, while task-present Macro IoU increased from 0.296 to 0.324. Natural-Language Routing also improved, with Macro Target IoU increasing from 0.286 to 0.302. In contrast, task-ID conditioning remained nearly unchanged, with Macro Target IoU changing from 0.269 to 0.268.

These results suggest that task conditioning becomes more useful when the model can adjust both which tokens are retained and how many tokens are needed for each image–task pair. A fixed retention ratio assumes that every input requires approximately the same amount of computation. This may be unnecessarily restrictive for semantic segmentation because some requested classes may occupy large or visually complex regions, while others may be small, absent, or relatively easy to identify. Adaptive retention provides the router with additional flexibility to respond to these differences.

The improvement from learned task embeddings is important because it changes the ordering observed under fixed retention. Natural-Language Routing was the strongest fixed task-conditioning strategy, but learned task embeddings achieved the highest Macro Target IoU after adaptive retention was introduced. At the same time, adaptive Natural-Language Routing achieved the highest semantic mIoU, Boundary IoU, and pixel accuracy. This demonstrates that there was not a clear winner across every metric, since learned embeddings performed best on the primary class-conditioned overlap metrics, while Natural-Language Routing performed better on several broader segmentation metrics.

The adaptive improvements also occurred without a large increase in the average number of retained tokens. Learned task embeddings and Natural-Language Routing retained approximately 76.5% of Stage-2 tokens, compared with the fixed 75% target. This difference is relatively small compared with the improvement in Macro Target IoU, particularly for learned task embeddings. These results suggest that adaptive token allocation can improve performance while maintaining an average retention ratio close to the fixed-retention budget. However, because the adaptive models retained slightly more tokens on average, the improvement cannot only be attributed to adaptive allocation.

However, adaptive routing still did not exceed the strongest task-agnostic method on the primary metric. Adaptive learned embeddings achieved a Macro Target IoU of 0.307 compared with 0.312 for random pruning. The adaptive experiment therefore closes much of the performance gap observed under fixed routing, but the current results still do not demonstrate an overall advantage over simple task-agnostic token reduction.

6.4. Overall Findings

Overall, the results show that task-agnostic and task-conditioned token reduction produce different accuracy–efficiency tradeoffs depending on the routing strategy and whether retention is fixed or adaptive. Random pruning achieved the highest observed Macro Target IoU among the reduced-token configurations, showing that a simple task-agnostic strategy can preserve segmentation performance under moderate token reduction. The performance of random and visual-importance pruning also demonstrates that task conditioning is not necessary to obtain competitive segmentation performance at the tested token budget.

Within the task-conditioned methods, the choice of task representation clearly affected performance. Natural-Language Routing was the strongest strategy under fixed retention, while learned task embeddings produced the strongest Macro Target IoU when combined with adaptive retention. The improvement produced by adaptive routing suggests that determining how much computation should be allocated to an input may be just as important as determining which tokens should be retained.

The fixed and adaptive comparisons therefore provide a more complete answer to the research question than either experiment individually. Under fixed retention, the task-agnostic strategies achieved higher observed Macro Target IoU scores than the task-conditioned strategies. When adaptive retention is introduced, the task-conditioned models become considerably more competitive, particularly learned task embeddings, although random pruning still achieved the highest observed score on the primary metric. These results suggest that task information may become more useful when the routing mechanism is given additional flexibility over the amount of computation allocated to each image–task pair.

Based on these results, adaptive learned task embeddings provide the strongest task-conditioned configuration for the primary evaluation metric, while adaptive Natural-Language Routing remains useful because of its strong semantic and boundary performance and its ability to accept alternative task descriptions. Random pruning achieved the highest observed Macro Target IoU among the reduced-token strategies and should therefore remain an important comparison for future task-conditioned routing methods.

6.5. Limitations

This study has several limitations. First, the experiments are conducted on the LoveDA remote-sensing

dataset using a lightweight PVTv2-style backbone and a single routing location. Token reduction is applied at Stage 2, and the adaptive method predicts a single retention ratio within the specified range. As a result, the findings may not directly generalize to larger transformer architectures, different routing stages, higher-capacity segmentation models, or domains beyond remote sensing. LoveDA also contains class imbalance and domain-specific urban and rural visual patterns, which may affect how reliably the observed routing behavior transfers to datasets with different object scales, class distributions, or boundary complexity.

Additionally, the routing methods use dense reconstruction and masking instead of true sparse execution. As a result, the reported computational reductions represent analytical attention savings and not equivalent measured reductions in latency, memory use, or wall-clock inference time. A deployment-oriented implementation would be required to determine whether the reduction in effective token computation translates into practical hardware-level speedups.

Natural-Language Routing is also intentionally lightweight. The bag-of-words prompt features allow alternative descriptions of the known segmentation classes to be evaluated, but they do not provide the semantic representation expected from a pretrained vision-language model. The prompt-transfer experiment also evaluates alternative descriptions of known classes rather than completely unseen semantic tasks. Therefore, the prompt-transfer results should not be interpreted as evidence of open-vocabulary segmentation.

6.6. Future Research

Future research could extend the comparison of task-agnostic and task-conditioned token reduction in several directions. One important next step would be implementing true sparse token execution so that the analytical reduction in attention computation can be compared with measured inference latency, memory consumption, throughput, and hardware utilization. This would provide a more complete evaluation of whether reductions in effective token computation translate into practical efficiency improvements.

The task representations could also be expanded beyond the approaches used in this study. In particular, stronger pretrained language or vision-language representations, such as CLIP-based embeddings, could be used to determine whether richer semantic information improves token selection. This could also allow future experiments to evaluate whether stronger task-conditioned methods can outperform the simple task-agnostic strategies observed in this study while supporting generalization beyond alternative descriptions of known classes.

Adaptive routing could also be extended beyond predicting a single retention ratio at Stage 2. A model could dynamically determine both how many tokens should be retained and where token reduction should occur throughout the transformer, allowing different stages to allocate computation according to the requested task and complexity of the input. Evaluating these approaches with larger vision transformer architectures and additional segmentation datasets could help determine whether the differences between task-agnostic and task-conditioned reduction observed in this study remain consistent across different architectures, domains, and segmentation tasks.

7. CONCLUSION

This research compared task-agnostic and task-conditioned token reduction strategies to evaluate their accuracy–efficiency tradeoffs for vision transformer semantic segmentation. It also investigated whether adaptive retention improves upon fixed token retention. This framework allowed the effects of token reduction, task conditioning, and adaptive retention to be evaluated under a consistent segmentation objective while still measuring class-conditioned performance through Macro Target IoU.

The results show that token reduction can maintain segmentation performance even without task information. Random token pruning achieved performance comparable to the full-token baseline while retaining only 75% of Stage-2 tokens. This suggests that a portion of the intermediate visual tokens can be removed without severely affecting segmentation quality. Among the fixed task-conditioned approaches, Natural-language routing performed best, while learned task embeddings also performed well, and the lowest performing model was discrete task IDs. However, none of the fixed task-conditioned methods achieved a higher observed Macro Target IoU than random pruning. Therefore, the results do not demonstrate that task-conditioned routing is inherently better than simpler token reduction. However, they suggest that richer task representations can provide more useful routing signals within a task-conditioned framework.

The adaptive token-retention experiment changed the ordering of the task conditioned results. When the model was allowed to predict the retained-token ratio for each image task pair, learned task embeddings achieved the strongest task-conditioned Macro Target IoU, while adaptive Natural-Language Routing achieved the strongest semantic mIoU and Boundary IoU among the task-conditioned models. This suggests

that adaptive retention can make task conditioning more useful by allowing the model to adjust both which tokens are retained and how many tokens are needed for a given input. However, adaptive routing still did not achieve a higher observed Macro Target IoU than random pruning.

This study overall demonstrates that token reduction can be applied without severely degrading segmentation performance. It also shows that task-conditioned routing is most effective when combined with adaptive retention, especially for learned task embeddings. At the same time, the results show the importance of comparing more complex routing methods against strong and simple token-reduction baselines, since random pruning achieved the highest observed Macro Target IoU among the reduced-token methods in these experiments. More improvements to the routing mechanism, richer prompt representations, adaptive token retention, and true sparse execution could help determine whether task-conditioned routing can actually provide stronger accuracy–efficiency tradeoffs for vision transformers and become more practical for applications with computational limits.

8. REFERENCES

[1] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 568–578.

[2] D.-J. Lee, J. Hur, J. Choi, J. Yu, and J. Kim, "Frequency-aware token reduction for efficient vision transformer," in *Advances in Neural Information Processing Systems*, vol. 38, 2025.

[3] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C.-J. Hsieh, "Dynamicvit: Efficient vision transformers with dynamic token sparsification," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 21 329–21 342.

[4] Y. Liang, C. Ge, Z. Tong, Y. Song, J. Wang, and P. Xie, "Not all patches are what you need: Expediting vision transformers via token reorganizations," in *International Conference on Learning Representations*, 2022.

[5] D. Bolya, C.-Y. Fu, X. Dai, P. Zhang, and C. Feichtenhofer, "Token merging: Your vit but faster," in *International Conference on Learning Representations*, 2023.

[6] H. Chen et al., "Vltp: Vision-language guided token pruning for task-oriented segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025, pp. 9335–9345.

[7] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proceedings of the International Conference on Machine Learning*, 2021, pp. 8748–8763.

[8] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.