BUDGETED VISUAL TOKEN REFRESH FOR MULTI-TURN LARGE VISION–LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual-token pruning is widely used to reduce the inference cost of multimodal large language models by retaining only a subset of visual tokens. In multi-turn dialogue, however, reusing a visual-token set selected for the first question can make later questions unable to access relevant visual evidence even when that evidence is still present in the image. We formulate this failure mode as the problem of revising the active visual context during dialogue under a fixed token budget, and propose Budgeted Visual Token Refresh (BVTR). BVTR removes a bounded number of currently active tokens and reactivates previously inactive tokens according to their relevance to each follow-up question, while keeping the total number of active visual tokens fixed. To address the context inconsistency introduced by token replacement, we use dense reconstruction as a consistent reference and additionally evaluate a selective sparse state-update implementation. On the controlled COCO-SR1000 evaluation with LLaVA-1.5-7B, BVTR-Sparse96 improves accuracy from 88.9% with Fixed-288 to 91.0% under the same 288-token budget, closing 95.5% of the observed Fixed-to-Full accuracy gap relative to Full-576 at 91.1%. On COCO-MT4-200, the fraction of dialogues in which all three follow-up questions are answered correctly increases from 68.0% to 75.5%, indicating that the improvement persists under repeated shifts of visual focus. On Qwen3-VL-8B-Instruct, dense BVTR under the same 50% visual-token budget improves accuracy by 2.1 percentage points over Fixed-50%. By contrast, gains are limited on VQAv2-Shift500 and when starting from a strong initial selector. These results suggest that an important question in multi-turn visual-token pruning is not only what to discard, but how to reallocate a limited active visual budget as the user's question changes.

1 INTRODUCTION

Large Vision–Language Models (LVLMs) jointly process images and natural language and have achieved strong performance on visual question answering, image understanding, and multi-turn dialogue. Images, however, are represented by hundreds of visual tokens, which introduce substantial computation and memory overhead during inference. A growing line of work therefore reduces this cost by pruning low-importance visual tokens and retaining only a subset of them.

Most visual-token pruning methods focus on selecting the tokens that are most useful for the current query. This is effective for a single question, but it can omit information that becomes necessary later in a multi-turn dialogue. If a visual-token set selected for the first question is reused throughout the conversation, a later shift in user focus may require visual evidence that was removed by the first-turn pruning decision. The information still exists in the image, yet it is no longer available in the active context. Because user focus changes over time, the visual-token selection need not remain permanently tied to the first question.

We therefore view visual-token pruning as an update problem: under a limited budget, which visual information should remain active at the current turn? We propose Budgeted Visual Token Refresh (BVTR), which revises the active visual context after each follow-up question while keeping the active-token budget fixed. In the default LLaVA-1.5 setting, BVTR activates 288 of 576 visual tokens for the first question and replaces at most 48 of them at each later turn according to their relevance to

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.

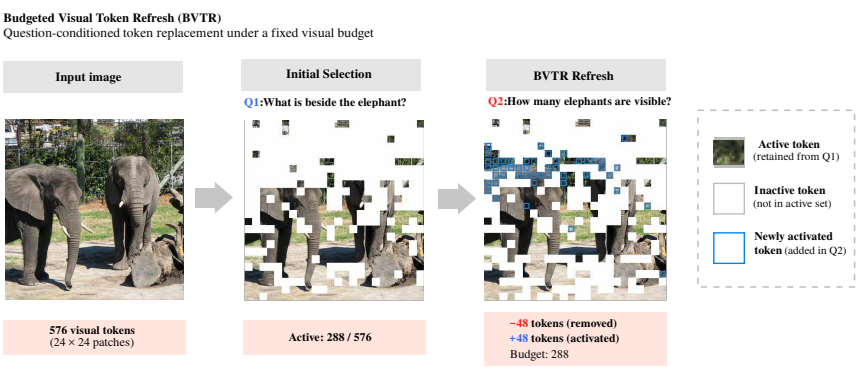

Figure 1: Overview of Budgeted Visual Token Refresh (BVTR). The first question activates 288 of 576 visual tokens. When the visual focus changes at a follow-up turn, BVTR replaces a bounded subset of the active tokens while preserving the same visual-token budget. Blue outlines indicate tokens newly activated for the follow-up question.

the current question. The key point is that BVTR does not keep adding tokens or increase the active budget; it changes the composition of the same 288-token budget.

Token replacement alone is not sufficient. The KV states of newly activated tokens and the states already stored in the cache were computed under different visual contexts. Naively inserting only the new states can therefore mix contextually inconsistent states within a single cache. The most direct solution is dense reconstruction, which recomputes the updated 288-token visual context together with the prior dialogue states across all layers; we use this as a consistency reference. We additionally evaluate a sparse state-update implementation that selectively recomputes only part of the cached state to reduce reconstruction cost. The central contribution of this work is therefore not the particular Sparse96 configuration itself, but the bounded-refresh principle: first-turn visual selection is revisable rather than permanent.

Our evaluation asks whether BVTR recovers visual evidence omitted by static pruning and whether this advantage persists under repeated changes of focus. COCO-SR1000 is a controlled diagnostic designed to test whether BVTR can recover follow-up evidence excluded by the first-turn static selection. COCO-MT4-200 evaluates whether the same update mechanism remains useful across repeated visual-focus shifts. We further test whether the direction of the bounded-refresh effect transfers to Qwen3-VL-8B-Instruct and examine boundary conditions on VQAv2-Shift500 and a TransPrune-based evaluation.

Our contributions are:

- We formulate multi-turn visual-token pruning not as a one-time decision about what to permanently discard, but as a budgeted refresh problem in which the active visual set can be revised as follow-up questions change.
- We introduce question-conditioned bounded replacement, which swaps a limited number of active tokens while preserving a fixed visual-token budget, and compare dense reconstruction with a selective sparse state-update implementation for handling the resulting context inconsistency.
- Through controlled omission recovery, repeated multi-turn shifts, cross-model validation, and boundary-condition analysis, we systematically characterize when BVTR helps and when its additional benefit is limited.

2 Related Work

2.1 Visual-Token Pruning

LVLMs inject a large number of visual tokens produced by a vision encoder into the language-model context. These tokens preserve fine-grained image information, but increase self-attention cost and KV-cache memory. Prior work reduces this overhead using attention scores, token similarity, spatial redundancy, or question relevance to identify and remove less useful visual tokens (Chen et al., 2024; Zhang et al., 2025; Xing et al., 2025; Wang et al., 2026; Li et al., 2026).

Early approaches such as FastV perform token selection at shallow layers and forward only the selected subset to deeper layers, while later methods progressively reduce tokens or improve selection through richer token-transition and compression strategies (Chen et al., 2024; Xing et al., 2025; Li et al., 2026). Recent work has also begun to study token reduction explicitly for multi-turn VQA. In particular, Rethinking Token Reduction for Large Vision-Language Models explores prompt-agnostic compression intended to preserve information for unknown future queries (Wang et al., 2026). BVTR takes a complementary direction: instead of requiring a single compressed visual representation to serve all later questions, it keeps the fixed-size active visual context revisable and reallocates that budget after each follow-up question arrives.

BVTR is thus a refresh framework that operates above a given token selector. Its main distinction is not a new first-turn pruning score, but how the selected active set is revised during dialogue under a fixed token budget.

2.2 KV Cache Reuse and Context Update

Autoregressive Transformers store the keys and values of previous tokens in a KV cache to avoid repeatedly recomputing the same context. However, the state of each layer depends not only on the token itself but also on the preceding context from which it was computed. If the visual-token set changes and only the KV states of newly activated tokens are inserted into the existing cache, states produced under different visual contexts can become mixed.

The safest solution is to reconstruct the context using the updated visual-token set. Dense reconstruction, however, requires reprocessing the active visual tokens at every follow-up turn. At the other extreme, simply reusing the old cache is inexpensive but may fail to reflect the updated visual context. Recent work on editable and composable KV caches further demonstrates that cached states can encode conclusions conditioned on earlier context rather than behaving as independent per-token records (Li, 2026). Motivated by this context dependence, BVTR evaluates selective contextual repair between the two extremes.

2.3 Multi-Turn Vision–Language Reasoning

BVTR targets dialogues in which the user's visual focus shifts while the image remains fixed. The current implementation evaluates all candidate visual tokens up to an intermediate layer to obtain question-conditioned relevance, but does not fully reconstruct the updated visual context across every decoder layer at each turn. Instead, it revises the contents of a bounded active set and, in the sparse implementation, repairs only a selected subset of deeper-layer states.

3 Problem Formulation

Let an image I produce N visual tokens:

$$V = \{v_1, v_2, \ldots, v_N\}. \tag{1}$$

Let q_t denote the question at turn t and H_{t-1} the preceding dialogue history. The model uses only B active visual tokens, with active index set

$$S_t \subseteq \{1, \ldots, N\}, \qquad |S_t| = B. \tag{2}$$

In the default LLaVA-1.5 setting, $N = 576$ and $B = 288$.

Static pruning reuses the set selected at the first turn:

$$S_t = S_1, \qquad t > 1. \tag{3}$$

If a later question q_t requires an image region that was not retained in S_1, the necessary token may no longer be available in the active context. BVTR instead chooses a set D_t to remove from the previous active set and a set A_t to activate:

$$S_t = (S_{t-1} \setminus D_t) \cup A_t, \qquad |D_t| = |A_t| \leq M. \tag{4}$$

Here M is the maximum number of replacements per turn; we use $M = 48$ by default. The active-token count therefore remains fixed at B even as the question changes.

The objective is to reduce follow-up prediction loss under this fixed-budget and bounded-change constraint:

$$\min_{\{S_t\}} \mathbb{E}[\mathcal{L}(y_t, \hat{y}_t)] \quad \text{s.t.} \quad |S_t| = B,\ |S_t \setminus S_{t-1}| \leq M. \tag{5}$$

Here y_t and $\hat{y}_t$ denote the target answer and model prediction. In practice, this accuracy objective must be considered together with refresh latency and incremental peak memory.

4 Budgeted Visual Token Refresh

4.1 Question-Conditioned Bounded Replacement

Let $s_t(i)$ denote the importance of visual token v_i for follow-up question q_t. Our default implementation uses an attention-based relevance score between question tokens and visual tokens at an intermediate Transformer layer, without using answers, object bounding boxes, or evaluation labels.

High-scoring tokens outside the previous active set become activation candidates, while low-scoring active tokens become removal candidates. BVTR replaces equally sized subsets, up to M tokens:

$$(A_t, D_t) = \mathrm{Refresh}(s_t, S_{t-1}, M). \tag{6}$$

Ties are resolved by the original patch index. This bounded update differs from global reselection, which selects a new set of B tokens from all N candidates at every turn. BVTR retains most of the previous active context and introduces only a limited amount of newly relevant visual information.

4.2 Context Inconsistency after Replacement

When the active set changes from S_{t-1} to S_t, the existing KV cache is no longer fully consistent with the new visual context. The deeper states of retained visual tokens, as well as prior question and answer states, were already computed using the previous active set. Inserting only the KV states of newly selected tokens would therefore mix states produced from different visual contexts in the same cache.

This issue is not solved by position correction alone. Even if each visual token preserves its original rasterized visual-sequence position, the context under which an existing state was computed remains unchanged. After token replacement, internal states must therefore be updated to reflect the new active set.

4.3 Dense Reconstruction as a Reference

BVTR-Dense uses the updated active set S_t to replay the previous question and the token IDs of the answer that was actually generated, reconstructing states under the new visual context. This consistently reflects the updated active set across all Transformer layers, but requires recomputing the B active visual states through the full network. We therefore use Dense as a consistency reference after bounded refresh and compare it against selective sparse state updates that reduce the amount of recomputation.

4.4 Sparse Contextual Repair

BVTR-Sparse selectively updates only part of the state after token replacement. Newly activated tokens require states consistent with the updated visual context and are therefore recomputed, while the repair scope for retained tokens is controlled by a separate repair budget.

BVTR-Sparse first processes all B updated active tokens through the first E layers. For a token i retained across consecutive turns, we define its KV change at layer E as

$$d_t(i) = \|k_{t,i}^{(E)} - k_{t-1,i}^{(E)}\|_2 + \|v_{t,i}^{(E)} - v_{t-1,i}^{(E)}\|_2. \tag{7}$$

Given repair budget K, the repair set is

$$P_t = A_t \cup \mathrm{TopK}_{K-|A_t|}\left(\{d_t(i) : i \in S_t \cap S_{t-1}\}\right). \tag{8}$$

The frozen BVTR-Sparse96 configuration used in the confirmatory evaluation sets $E = 2$ and $K = 96$. At deeper layers, only states in P_t are recomputed; the remaining states are reused from the previous cache. Recomputed and reused states are merged according to their original patch indices so that token order and logical positions are preserved. In the post-hoc ablation, settings from $K = 48$ through dense reconstruction achieve the same observed accuracy, so we do not claim that either $K = 96$ or high-drift retained-token selection is necessary for the accuracy gain.

For a Transformer with L layers, the approximate number of token-layer state updates is

$$U_{\mathrm{dense}} = LB, \tag{9}$$

$$U_{\mathrm{sparse}} = EB + (L - E)K. \tag{10}$$

This is not an exact FLOP count; it only compares the number of states updated. End-to-end latency additionally includes question-conditioned selection, answer-token replay, cache merging, and GPU kernel efficiency, and is therefore measured separately.

5 Experiments

5.1 Experimental Setup

5.1.1 Research Questions

Our experiments are organized around five questions:

- RQ1: Under the same active-token budget, can BVTR recover follow-up information omitted by static pruning?
- RQ2: Does the improvement persist under repeated shifts of visual focus across multiple turns?
- RQ3: How much state updating is required after bounded replacement, and what accuracy–cost trade-off does selective repair provide relative to dense reconstruction?
- RQ4: Does the direction of the bounded-refresh effect transfer to another LVLM?
- RQ5: Under which question formats and initial-selection conditions is the additional benefit of refresh limited?

5.1.2 Models and Data

Our main experiments use LLaVA-1.5-7B. The default configuration contains 576 visual tokens, an active budget of 288, and a maximum of 48 replacements per turn. The frozen Sparse configuration uses $E = 2$ initial dense layers and a repair budget of $K = 96$.

COCO-SR1000 is a controlled diagnostic evaluation containing 1,000 distinct images, with 500 positive and 500 negative follow-up questions. It is designed to measure whether BVTR can recover evidence that becomes necessary after being excluded by the first-turn static selection. COCO-MT4-200 contains four questions about different targets per image and repeatedly applies the same refresh rule over three shifts after the first question.

Table 1: Controlled diagnostic results on COCO-SR1000. All 288-token conditions use the same active-token budget.

Method	Accuracy (%)	Δ vs. Fixed (%p)
Fixed-288	88.9	–
Random Refresh-288	89.3	+0.4
Previous-Question Refresh-288	89.1	+0.2
Global Reselection-288	90.9	+2.0
BVTR-Dense Reference	90.8	+1.9
BVTR-Sparse96	**91.0**	**+2.1**
Full-576	91.1	+2.2

For broader open-ended VQA validation, we use VQAv2-Shift500, containing 500 distinct images and 1,000 non-overlapping questions. Follow-up answer types are balanced across 167 yes/no, 167 number, and 166 other examples, and model outputs are not used to construct the dataset. For cross-model validation, we use Qwen3-VL-8B-Instruct on the same COCO-SR1000 images while activating 50% of the model-native visual tokens for each image.

5.1.3 Baselines and Metrics

The main conditions are Fixed-288, Random Refresh, Previous-Question Refresh, Global Reselection-288, BVTR-Dense Reference, BVTR-Sparse96, and Full-576. All methods are evaluated with the same question order and evaluation budget.

For binary questions, we report accuracy, precision, recall, specificity, and F1, together with an exact two-sided McNemar test over paired correctness outcomes for the same images. On MT4, we additionally report sequence success, defined as answering all three follow-up questions correctly. For VQAv2, we use the official 10-answer consensus score, a paired bootstrap 95% confidence interval, and a two-sided paired sign-flip test. Efficiency is evaluated using follow-up-turn end-to-end latency, including selection and replay where applicable, together with incremental peak memory measured relative to the live state at the start of each condition.

5.2 Controlled Diagnosis: COCO-SR1000

BVTR-Sparse96 improves accuracy by 2.1 percentage points over Fixed-288. It recovers 23 errors made by Fixed while harming 2 previously correct cases, yielding a significant paired difference ($p = 1.94 \times 10^{-5}$). Random Refresh and Previous-Question Refresh improve by only 0.4 and 0.2 percentage points, respectively. This indicates that the effect cannot be explained by arbitrary token exchange alone and instead depends on selecting tokens relevant to the current question.

BVTR-Sparse96 is 0.2 percentage points above the Dense reference and 0.1 points below Full-576 in observed accuracy. We do not interpret these small differences as evidence that Sparse is intrinsically more accurate than Dense or equivalent to Full. Rather, the controlled result shows that BVTR can recover follow-up visual evidence omitted by static pruning under the same active-token budget.

5.3 Repeated Multi-Turn Shifts: COCO-MT4-200

On COCO-MT4-200, sequence success across all three follow-up turns increases from 68.0% (136/200) with Fixed-288 to 75.5% (151/200) with BVTR-Sparse96. Sparse96 succeeds on 17 dialogues where Fixed fails, whereas Fixed succeeds on only 2 dialogues where Sparse96 fails; the exact McNemar test is significant ($p = 7.29 \times 10^{-4}$). This shows that the omission-recovery effect observed for a single Q1–Q2 shift can persist across repeated changes of visual focus.

5.4 Efficiency and Ablations

Sparse repair is intended to preserve the accuracy recovery from bounded replacement while reducing the amount of state recomputation required by Dense reconstruction. For a fair comparison, we remeasure follow-up-turn end-to-end latency for Fixed-288, BVTR-Dense, BVTR-Sparse96, and

Table 2: Per-turn accuracy and sequence success on COCO-MT4-200. Each condition evaluates the same 200 dialogues.

Method	T2 (%)	T3 (%)	T4 (%)	All T2–T4 (%)
Fixed-288	89.0	86.5	89.0	68.0
BVTR-Dense	92.5	90.5	92.5	75.5
BVTR-Sparse96	**93.0**	90.0	**92.5**	75.5
Full-576	**93.0**	90.0	**92.5**	**76.0**

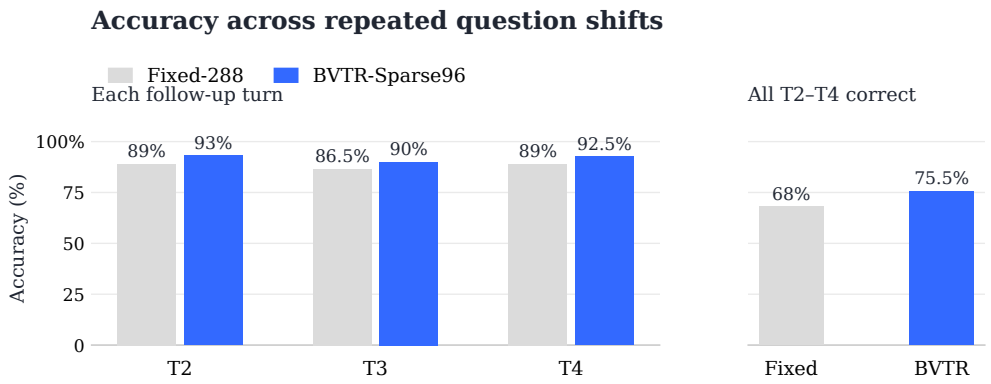

Figure 2: Accuracy across repeated question shifts on COCO-MT4-200. BVTR-Sparse96 remains above Fixed-288 at each follow-up turn and increases sequence success across all three follow-up questions from 68.0% to 75.5%.

Full-576 on 100 COCO-SR1000 examples using the same timing protocol and batched replay for paths that reconstruct prior answer states.

BVTR-Sparse96 has a median end-to-end latency of 296.2 ms, compared with 354.4 ms for BVTR-Dense. In paired per-example comparisons, Sparse96 reduces latency by a median of 19.9% relative to Dense reconstruction. Full-576 has a median latency of 335.9 ms, while Fixed-288 is the fastest at 143.7 ms. Sparse96 therefore reduces the update cost relative to Dense reconstruction, but is not faster than static pruning. We interpret this as an accuracy–efficiency trade-off: additional computation is used to recover accuracy lost when a static active set becomes misaligned with a later question. In a separate incremental-memory measurement, BVTR-Sparse96 requires 35.4% less additional peak allocation than Full-576.

In the post-hoc Ablation200 subset, observed accuracy increases from 90.0% at $M = 0$ to 92.5% at $M = 16$ and remains at 92.5% for $M = 32$, 48, 96, and global reselection. Likewise, all repair settings from $K = 48$ through Dense reconstruction yield the same observed accuracy of 92.5%. We therefore do not claim that $M = 48$, $K = 96$, or high-drift retained-token selection is necessary for the accuracy gain. Detailed results are provided in Appendix A.

5.5 Cross-Model Validation: Qwen3-VL-8B

For Qwen3-VL-8B-Instruct, we activate 50% of the model-native visual tokens for each image and compare Fixed-50%, BVTR-Dense-50%, and Full-100%. We retain Qwen3's native 3-axis mRoPE and DeepStack mechanisms and do not reuse LLaVA-specific cache or position-handling code. This experiment tests whether the accuracy effect of bounded dense refresh transfers across model families; it does not claim cross-model generalization of Sparse96.

BVTR-Dense-50% improves accuracy by 2.1 percentage points over Fixed-50%. It recovers 24 Fixed errors while harming 3 previously correct cases, producing a significant paired difference ($p = 4.92 \times 10^{-5}$). At the same observed accuracy of 96.0% as Full-100%, its incremental peak memory is 43.2% lower. Its end-to-end latency is nevertheless higher than Full-100% because of

Table 3: Latency benchmark on 100 COCO-SR1000 examples under a common batched-replay timing protocol. Accuracy values are from the full evaluation.

Method	Accuracy (%)	Median follow-up E2E (ms)
Fixed-288	88.9	**143.7**
BVTR-Dense	90.8	354.4
BVTR-Sparse96	91.0	296.2
Full-576	91.1	335.9

Table 4: Cross-model results on Qwen3-VL-8B-Instruct.

Method	Accuracy (%)	E2E (ms)	Inc. peak (MiB)
Fixed-50%	93.9	478.3	87.4
BVTR-Dense-50%	**96.0**	741.1	87.4
Full-100%	**96.0**	586.4	154.1

selector cost. The result therefore supports transfer of the bounded-refresh accuracy effect beyond LLaVA, but does not imply superiority on every efficiency metric.

5.6 Boundary Conditions

On VQAv2-Shift500, BVTR-Sparse96 improves over Fixed by only 0.34 percentage points, and the difference is not statistically significant ($p = 0.570$). The TransPrune-based evaluation likewise shows no additional improvement from BVTR ($p = 0.6875$). These results suggest that refresh offers limited additional benefit when the active set selected for the first question already preserves the visual evidence required by later questions. Detailed analyses are given in Appendix B.

6 Limitations

First, BVTR requires additional computation for question-conditioned token selection and state updating. In the current implementation, BVTR-Sparse96 requires 2.06× the follow-up-turn latency of Fixed-288, so we do not claim unconditional latency superiority over static visual-token pruning. Reducing question-conditioned selection cost remains an important direction for future work.

Second, the sparse implementation is primarily validated on LLaVA-1.5-7B. The Qwen3-VL-8B experiment confirms the direction of the accuracy effect for dense bounded refresh, but does not establish cross-model generality for Sparse96 itself.

Third, the benefit of bounded refresh is not uniform across settings. The limited gains on VQAv2-Shift500 and in the TransPrune-based evaluation suggest that refresh becomes less necessary when the first-turn active set already contains sufficient evidence for later questions.

7 Conclusion

We study multi-turn visual-token pruning as the problem of reallocating a fixed visual budget over the course of a dialogue rather than making a permanent pruning decision at the first question. BVTR replaces a bounded number of visual tokens according to the current question. Dense reconstruction and selective sparse state updating are evaluated as two implementation paths for reflecting the updated active set in the model state.

On the controlled SR1000 diagnostic, BVTR recovers follow-up visual evidence omitted by static pruning, and the improvement persists under repeated shifts on MT4. The direction of the dense bounded-refresh effect also transfers to Qwen3-VL-8B. At the same time, VQAv2 and TransPrune-based evaluations expose settings in which additional refresh provides little benefit because the relevant information is already retained. Together, these results suggest that a central question in multi-turn visual-token pruning is not only "what should be permanently removed?" but also "how should a fixed visual budget be reallocated as the dialogue changes?"

Table 5: Boundary conditions for BVTR. VQAv2 uses the 10-answer consensus score, whereas the TransPrune-based evaluation uses exact-answer accuracy; the two scores should not be compared directly. The TransPrune condition uses one-stage 50% retention rather than the original multi-stage schedule.

Evaluation	Method	Score (%)
VQAv2-Shift500	Fixed-288	71.26
	BVTR-Dense	71.70
	BVTR-Sparse96	71.60
	Full-576	71.36
TransPrune-based SR1000	Static-288	**91.5**
	Global-288	91.3
	TransPrune+BVTR-288	91.3
	Full-576	91.1

AI USE STATEMENT

Generative AI tools were used in this work to assist with method implementation, experimental design and review, interpretation of experimental results, code development and revision, figure preparation, literature search and organization, drafting and editing parts of the manuscript, English translation of drafts and notes, organization of data and evaluation files, and revision of the paper title and wording.

Generative AI was not used to generate synthetic datasets or fabricate experimental results. All numerical results reported in the paper were computed from actual model executions and stored output files. Generative AI was also not used to produce mathematical claims or proofs for this work. All AI-assisted work was reviewed by the authors. The main experimental procedures and configurations were frozen before inspection of the final confirmatory results, and the major reported values were rechecked against individual experimental outputs and result records. AI-assisted code and experimental procedures were also verified through direct execution. The authors take responsibility for the final content of this work, including all text, claims, code, and visual materials produced with the assistance of generative AI.

ETHICS STATEMENT

All experiments use publicly available visual question answering and image datasets under their respective terms of use. The experiments reported in the final paper use COCO and VQAv2. This work does not involve human-subject research, and no new data containing personal information were collected. The purpose of this study is to analyze visual-token selection and inference efficiency in vision–language models; it is not intended to make judgments about individuals or to predict sensitive personal attributes.

REPRODUCIBILITY STATEMENT

The model configurations, visual-token budgets, refresh rules, dataset construction, and evaluation procedures are described in the methodology and experimental sections of the main paper. Additional implementation details, experimental environments, random seeds, dataset integrity checks, latency measurement procedures, and ablation results are provided in the appendix. The main experimental procedures, dataset construction, decision criteria, selector, replacement budget, and Sparse96 configuration were frozen before inspection of the final confirmatory results.

REFERENCES

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Computer Vision – ECCV 2024*, volume 15139 of *Lecture Notes in Computer Science*, pp. 19–35. Springer, 2024. doi: 10.1007/978-3-031-73004-7_2.

Ao Li, Yuxiang Duan, Jinghui Zhang, Congbo Ma, Yutong Xie, Gustavo Carneiro, Mohammad Yaqub, and Hu Wang. Transprune: Token transition pruning for efficient large vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39529–39538, June 2026.

Bojie Li. Models take notes at prefill: Kv cache can be editable and composable. *arXiv preprint arXiv:2606.17107*, 2026. doi: 10.48550/arXiv.2606.17107.

Yi Wang, Haofei Zhang, Qihan Huang, Anda Cao, Gongfan Fang, Wei Wang, Xuan Jin, Jie Song, Mingli Song, and Xinchao Wang. Rethinking token reduction for large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24727–24737, June 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*, pp. 14593–14603, June 2025.

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, 2025.

A Ablation Details

Ablation200 is a 200-example post-hoc subset drawn from COCO-SR1000 and is not an independent confirmatory evaluation. We use it to analyze the behavior of bounded replacement and sparse state updating.

A.1 Replacement Budget

Table 6: Accuracy on Ablation200 as a function of replacement budget M. All replacement-budget conditions use Dense reconstruction.

Condition	Accuracy (%)
Fixed: no replacement, no reconstruction	90.0
$M = 0$: no replacement, Dense	90.0
$M = 16$: Dense	92.5
$M = 32$: Dense	92.5
$M = 48$: Dense	92.5
$M = 96$: Dense	92.5
Global reselection: Dense	92.5

Observed accuracy increases from 90.0% at $M = 0$ to 92.5% at $M = 16$, and remains at 92.5% for $M = 32$, 48, 96, and global reselection. This subset therefore provides no evidence that $M = 48$ is better than $M = 16$, and we do not infer from this small post-hoc subset that a particular replacement budget is generally sufficient.

A.2 Repair Budget and Retained-Token Selection

All settings achieve the same observed accuracy of 92.5%. This ablation therefore does not show that high-drift retained-token selection or $K = 96$ is necessary for the accuracy gain. We nevertheless report Sparse96 because it was the implementation frozen before the confirmatory evaluation. The result leaves open the possibility that smaller or simpler state-update rules may be sufficient.

Table 7: Repair budget and retained-token selection on Ablation200 with $M = 48$. $K = 48$ updates only newly activated tokens at deeper layers. All conditions process all 288 active tokens through the first two layers.

Repair target	K	Accuracy (%)
Newly activated tokens only	48	92.5
New + highest-KV-change retained tokens	72	92.5
New + highest-KV-change retained tokens	96	92.5
New + random retained tokens	96	92.5
New + lowest-KV-change retained tokens	96	92.5
Dense reconstruction	288	92.5

B Detailed Boundary Conditions

B.1 VQAv2-Shift500

On VQAv2-Shift500, the VQA consensus scores for Fixed-288, BVTR-Dense, BVTR-Sparse96, and Full-576 are 71.26, 71.70, 71.60, and 71.36, respectively. Sparse96 exceeds Fixed by 0.34 percentage points, but the paired bootstrap 95% confidence interval includes zero and the paired sign-flip test is not significant ($p = 0.570$).

This result shows that BVTR does not universally improve open-ended VQA accuracy. Many VQAv2 question pairs may be answerable from the same global scene information, or the original 288 active tokens may already be sufficient. The need for refresh therefore depends on whether static selection actually removes visual evidence required by the follow-up question.

B.2 TransPrune-Based Evaluation

Using the official TransPrune scoring function in a one-stage 50% retention evaluation, Static-288, Global-288, TransPrune+BVTR-288, and Full-576 achieve accuracies of 91.5%, 91.3%, 91.3%, and 91.1%, respectively. TransPrune+BVTR is 0.2 percentage points below Static, with no significant difference ($p = 0.6875$). Global reselection and BVTR produce identical predictions on all 1,000 examples.

This evaluation is not a reproduction of the original TransPrune 7/9/12-layer 12.5% schedule. Instead, it is a scope test asking whether bounded refresh adds value on top of a strong initial set constructed using the official TransPrune score under one-stage 50% retention. The result suggests that additional refresh may be limited when the initial selector already preserves sufficient follow-up evidence or when bounded refresh and global reselection lead to the same predictions.

C Reproducibility Details

For the final follow-up-turn latency benchmark, all reported conditions were measured on the same GPU using synchronized CUDA timing. Fixed-288 reuses the existing first-turn cache and therefore requires no answer replay. Paths that reconstruct the visual context (BVTR-Dense, BVTR-Sparse96, and Full-576) replay the previously generated answer token IDs using batched replay. Each condition was measured three times per example after warmup, and the per-example median was used for aggregation. The benchmark contains 100 COCO-SR1000 examples.

- COCO-SR1000 SHA-256:
  `71d10f8a3c146e37afe74fb5a040fa44`
  `9fc605f35300eaf226a2a8dd7e96ca37`.
- VQAv2-Shift500 SHA-256:
  `6d3903ec2504e80983a72e55cb1a7e3`
  `c682c178a5295765a0d4d1b094b757805`.
- As an implementation sanity check for Qwen3-VL, the official full-model path and the manually reconstructed full-model path produced identical generated token IDs on all 3/3 checked examples.

Table 8: Implementation and evaluation environment used in the main experiments.

Item	Setting
Operating system	Ubuntu 22.04
GPU	NVIDIA RTX 3090, 24 GB
Available GPUs	2
Latency benchmark	Single GPU (`CUDA_VISIBLE_DEVICES=0`)
Python	3.10
PyTorch	2.7.1+cu118
Transformers	4.57.3
Main LVLM	LLaVA-1.5-7B
Cross-model LVLM	Qwen3-VL-8B-Instruct
Attention implementation	eager
Conversation template	`llava_v1`
Maximum new tokens	5
Active visual tokens	288 / 576
Maximum replacement M	48
Sparse early dense layers E	2
Sparse repair budget K	96
Confirmatory seeds	2026091801, 2026091802
Latency warmup runs	1
Latency repetitions	3

- The main protocol, datasets, decision criteria, selector, replacement budget, and Sparse96 repair configuration were frozen before inspection of the final confirmatory results.