Feather the Throttle: Revisiting Visual Token Pruning for Vision-Language Model Acceleration

Supplementary Material

A1. Results on Different Model Setup

We additionally experiment with using a DINOv2 + SigLIP visual encoder. As shown in Table A1, we observe the same behavior that removing RoPE substantially improves performance and incorporating uniform sampling is strong.

A2. Additional FEATHER Results

We compare **FEATHER** performance against FastV and PyramidDrop on all evaluated benchmarks in Table A2.

In addition, we show performance with respect to total runtime on a NVIDIA L40S in Figure A1.

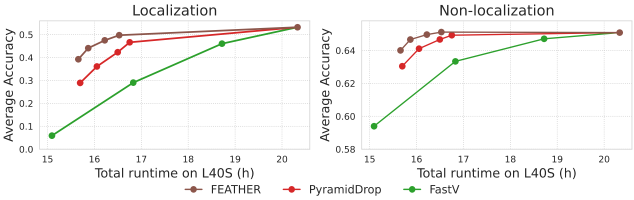


Figure A1. Total runtime on L40S vs. performance for FastV, PyramidDrop, and **FEATHER**.

Criteria	FLOPS Red	Localization					Open-Ended VQA					Challenge Sets				
		Avg	OCID-Ref	RefCOCOg	RefCOCO+	RefCOCO	Avg	TextVQA	GQA	VQAv2	VizWiz	Avg	POPE	TallyQA	VSR	AI2D
Attention-based																
ϕ_{original}	68%	27.2	21.9	27.7	27.8	31.1	56.6	35.6	59.1	74.0	57.7	66.1	84.6	60.2	**67.1**	52.7
$\phi_{\boldsymbol{-R}}$	68%	37.2	37.0	38.7	34.9	38.1	60.1	45.4	60.4	76.5	58.2	66.3	85.9	61.1	65.1	52.9
Δ		**+10.0**	**+15.0**	**+11.0**	**+7.0**	**+7.0**	**+3.5**	**+9.7**	**+1.2**	**+2.5**	**+0.5**	**+0.1**	**+1.3**	**+0.9**	**-2.0**	**+0.2**
Non-attention-based																
ϕ_{KNN}	66%	20.5	13.4	22.1	22.0	24.6	54.2	29.9	60.0	70.2	57.0	60.5	77.7	51.9	61.9	50.7
ϕ_{uniform}	66%	38.3	32.7	38.8	38.8	42.7	58.3	37.6	61.9	75.8	58.0	65.8	85.9	60.2	65.0	52.2
Ensemble																
$\phi_{\boldsymbol{-R}} + \phi_{\text{uniform}}$ (Ours)	**61%**	**46.3**	**41.6**	**47.3**	**46.0**	**50.1**	**61.3**	**46.8**	**62.0**	**77.7**	**58.7**	**66.8**	**86.9**	**61.6**	65.4	**53.3**

Table A1. Evaluating criteria using DINOv2 + SigLIP visual encoder. For each task, we **bold** the best result and underline the second-best result. Using $K = 3$ for all setups.

Method	FLOPS Red	GPU Hours	Localization					Open-Ended VQA					Challenge Sets				
			Avg	OCID-Ref	RefCOCOg	RefCOCO+	RefCOCO	Avg	TextVQA	GQA	VQAv2	VizWiz	Avg	POPE	TallyQA	VSR	AI2D
Baseline	0%	20.3	53.2	40.7	56.3	55.0	60.9	64.1	54.9	63.3	78.9	59.3	66.1	87.4	59.3	63.3	54.3
FastV	68%	15.1	5.9	5.7	5.1	6.1	6.7	54.8	31.8	58.4	72.7	56.3	64.0	83.2	57.1	63.3	52.4
PyramidDrop	65%	15.7	28.9	24.0	29.2	29.7	32.9	60.8	47.1	61.2	76.9	**57.9**	65.3	86.6	58.2	**63.4**	53.1
FEATHER	64%	15.7	**39.3**	**33.1**	**40.1**	**39.7**	**44.1**	**61.9**	**51.4**	**61.8**	**77.9**	56.5	**66.1**	**87.7**	**59.1**	**63.4**	**54.2**
FastV	45%	16.8	29.1	17.5	29.5	33.1	36.1	61.0	45.8	62.3	77.4	58.4	65.7	86.8	59.2	63.3	53.5
PyramidDrop	46%	16.8	46.6	37.4	48.3	47.8	53.0	63.7	53.8	63.1	78.7	**59.1**	66.2	87.5	**59.4**	63.5	54.3
FEATHER	48%	16.5	**49.7**	**39.3**	**52.1**	**50.9**	**56.7**	**63.9**	**54.6**	**63.2**	**78.8**	59.0	**66.3**	**87.7**	59.2	**64.0**	**54.6**

Table A2. Comparing **FEATHER** performance against FastV and PyramidDrop. The best results are **bolded** (excluding the baseline method).

A3. Comparison Against FasterVLM and VisionZip

We present FasterVLM and VisionZip performance in Table A3. We find that these approaches, while performing comparably to our approach on some benchmarks, perform vastly worse on localization benchmarks. We expect this is because positional information is not maintained in these methods, as image tokens are filtered without altering the positional embeddings. We verify the importance of positional embeddings in §A4. Note that since our setup uses the SigLIP encoder, for FasterVLM (which relies on [CLS] attention), we use the proposed solution in VisionZip of averaging attention each token receives from all others in the sequence.

A4. Token Shuffling Ablation

To assess the impact of positional embeddings on model performance, we shuffle positional embeddings for the image tokens and evaluate both the original VLM and our **FEATHER** approach. As shown in Table A3, the localization performance of both methods drops drastically for localization tasks, substantially for TextVQA, and relatively little for other benchmarks. This result supports our key insight that many vision-language benchmarks inadequately capture the shortcomings of efficiency methods due to their limited ability to assess fine-grained visual capabilities, particularly for visual grounding.

A5. Token Pruning Visualizations

In this supplemental material section, we provide a qualitative analysis comparing the pruning effectiveness of various criteria as well as the final approaches of **FEATHER**, FastV, and PyramidDrop. Namely, we visualize the ability of approaches to retain important tokens, particularly for localization. In Figure A2 and Figure A3, we visualize pruning from the various criteria assessed in the main text when pruning is done after layers three and eight, respectively. In Figure A4, we visualize pruning from the final approaches of **FEATHER**, FastV, and PyramidDrop.

A5.1. Comparing pruning criteria

We first visualize the retained tokens of various criteria when pruning is applied after layer three (see Figure A2) and layer eight (see Figure A3). We see that these visualizations support our quantitative results from the main paper. Specifically, (1) $\phi_{\text{-}\boldsymbol{R}}$ removes the criteria tendency of selecting bottom image tokens, resulting in an improved selection of maintained tokens; (2) the attention-based criteria improve when pruning after a later layer; and (3) adding uniform sampling to the attention-based pruning criteria with $\phi_{\text{-}\boldsymbol{R}} + \phi_{\text{uniform}}$ improves token selection.

		Localization					Open-Ended VQA					Challenge Sets				
Method	FLOPS Red	Avg	OCID-Ref	RefCOCOg	RefCOCO+	RefCOCO	Avg	TextVQA	GQA	VQAv2	VizWiz	Avg	POPE	TallyQA	VSR	AI2D
Baseline	0%	53.2	40.7	56.3	55.0	60.9	64.1	54.9	63.3	78.9	59.3	66.1	87.4	59.3	63.3	54.3
Baseline (pos shuffled)	0%	8.0	9.0	7.8	7.1	8.0	59.2	44.1	60.3	75.8	56.8	63.3	86.6	55.3	59.2	51.9
FasterVLM	65%	5.7	8.0	5.9	4.2	4.7	60.9	50.9	59.9	76.5	56.4	**66.6**	85.2	62.6	**63.7**	**54.7**
VisionZip	65%	8.5	7.3	9.0	8.1	9.5	61.1	50.8	60.2	76.7	**56.7**	66.5	85.3	**62.9**	**63.7**	54.3
FEATHER	64%	**39.3**	**33.1**	**40.1**	**39.7**	**44.1**	**61.9**	**51.4**	**61.8**	**77.9**	56.5	66.1	**87.7**	59.1	63.4	54.2
FEATHER (pos shuffled)	64%	5.3	5.3	4.8	5.2	5.8	57.8	41.7	58.9	75.0	55.5	63.2	86.0	55.7	58.8	52.5

Table A3. Comparison against FasterVLM and VisionZip and positional embeddings ablation (where image token positions are shuffled). The best results are **bolded**.

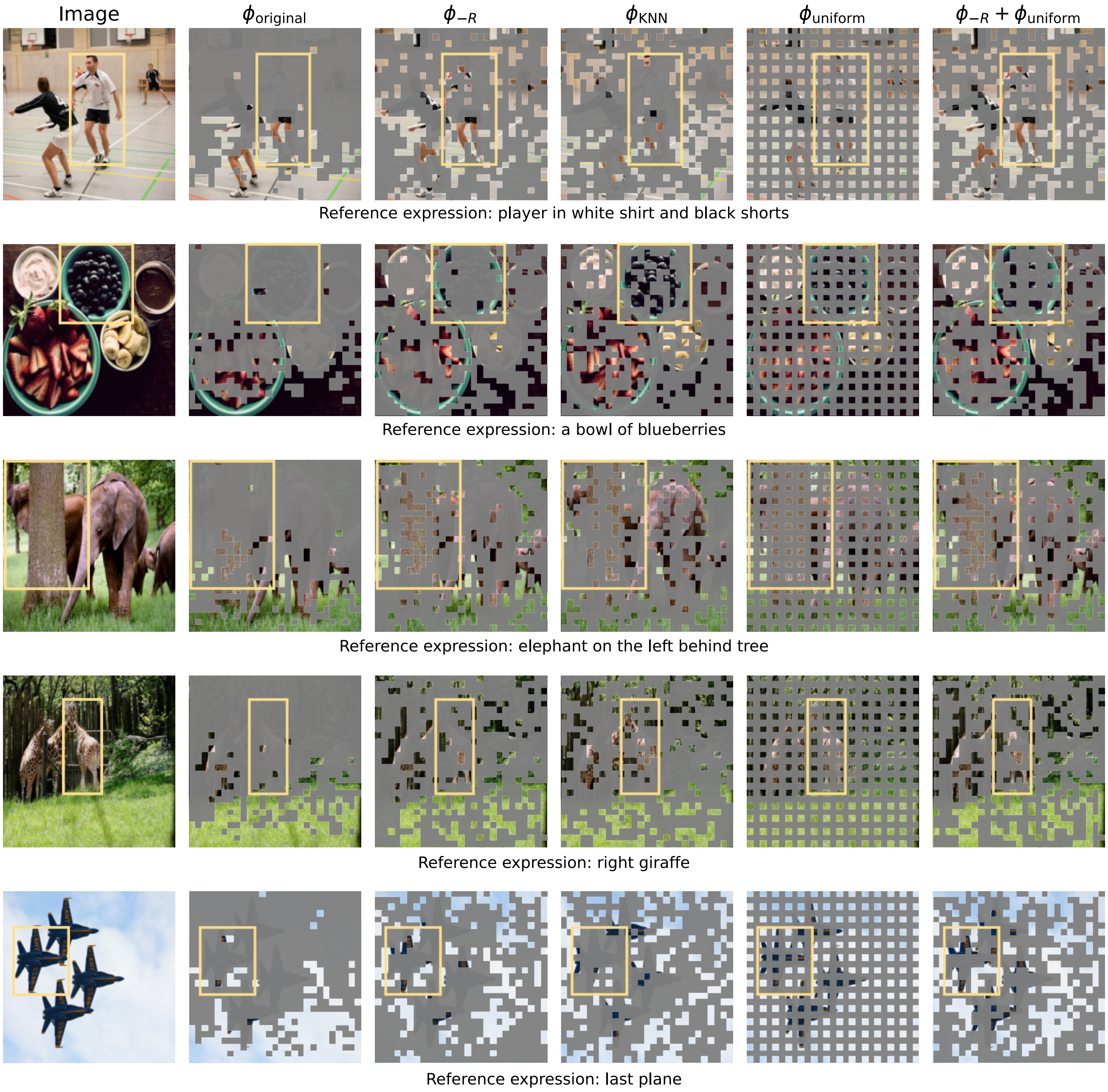
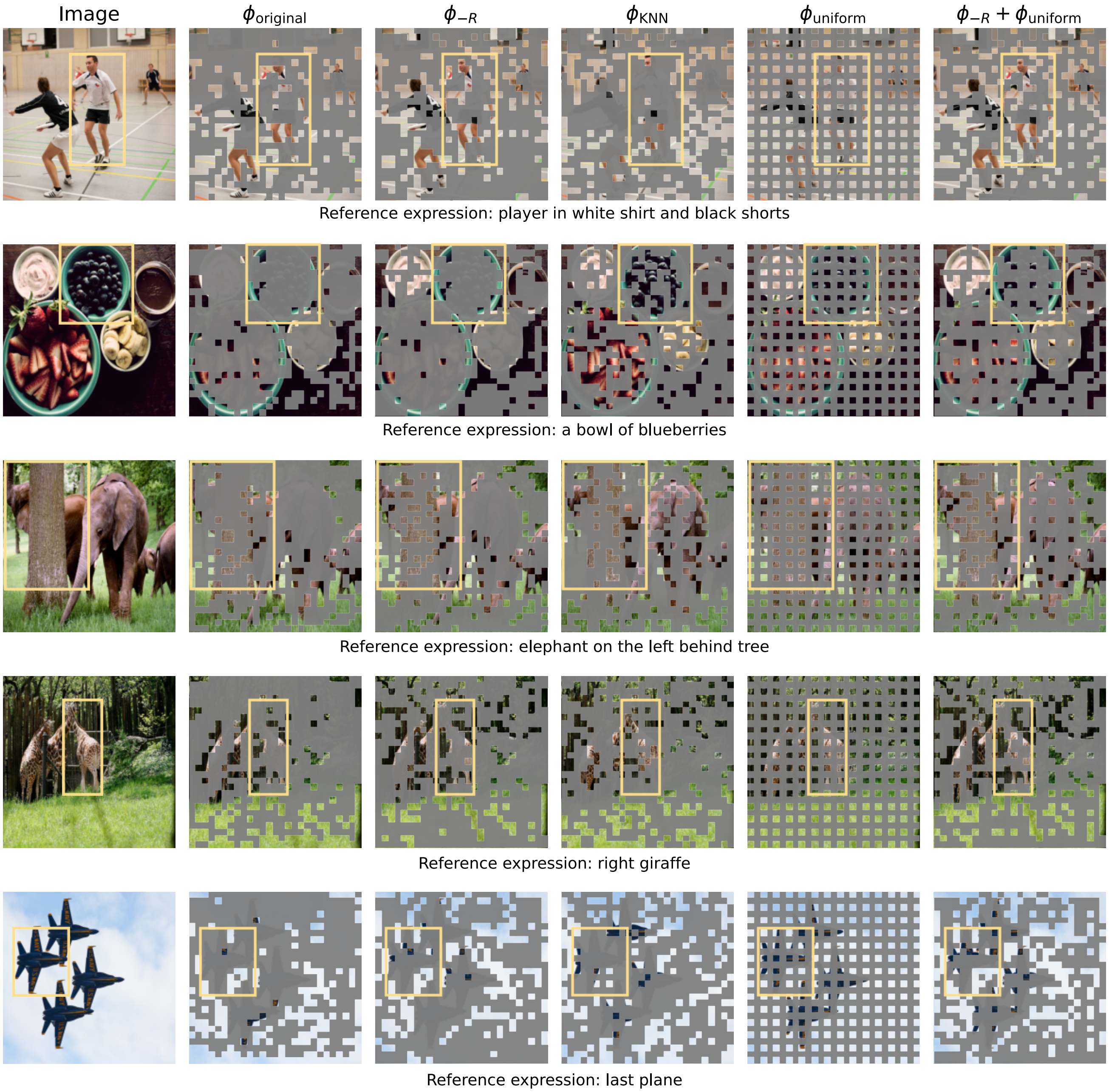


Image
ϕ_{original}
ϕ_{-R}
ϕ_{KNN}
ϕ_{uniform}
$\phi_{-R} + \phi_{\text{uniform}}$
Reference expression: player in white shirt and black shorts
Reference expression: a bowl of blueberries
Reference expression: elephant on the left behind tree
Reference expression: right giraffe
Reference expression: last plane

Figure A2. Visualizing the ability of various pruning criteria to maintain visual tokens relevant to the reference expression when applied after layer three. We observe that $\phi_{\text{-}\boldsymbol{R}}$ resolves ϕ_{original}'s tendency of selecting bottom image tokens and that uniform sampling is a robust approach that improves the token selection effectiveness of $\phi_{\text{-}\boldsymbol{R}}$ with $\phi_{\text{-}\boldsymbol{R}} + \phi_{\text{uniform}}$. See the main text for criteria definitions.



Reference expression: player in white shirt and black shorts

Reference expression: a bowl of blueberries

Reference expression: elephant on the left behind tree

Reference expression: right giraffe

Reference expression: last plane

Figure A3. Visualizing the ability of various pruning criteria to maintain visual tokens relevant to the reference expression when applied after layer eight. We observe that the attention-based criteria are more effective when pruning after this layer compared to after layer three. See the main text for criteria definitions.

A5.2. Comparing FEATHER to FastV and PyramidDrop

Additionally, we visualize the retained tokens for the FEATHER, FastV, and PyramidDrop approaches.

As shown in Figure A4, when comparing the remaining tokens used for prediction (after layer 16 for FEATHER, layer 24 for PyramidDrop, and layer three for FastV), we see that our approach retains substantially more tokens around and inside the reference expression bounding box.

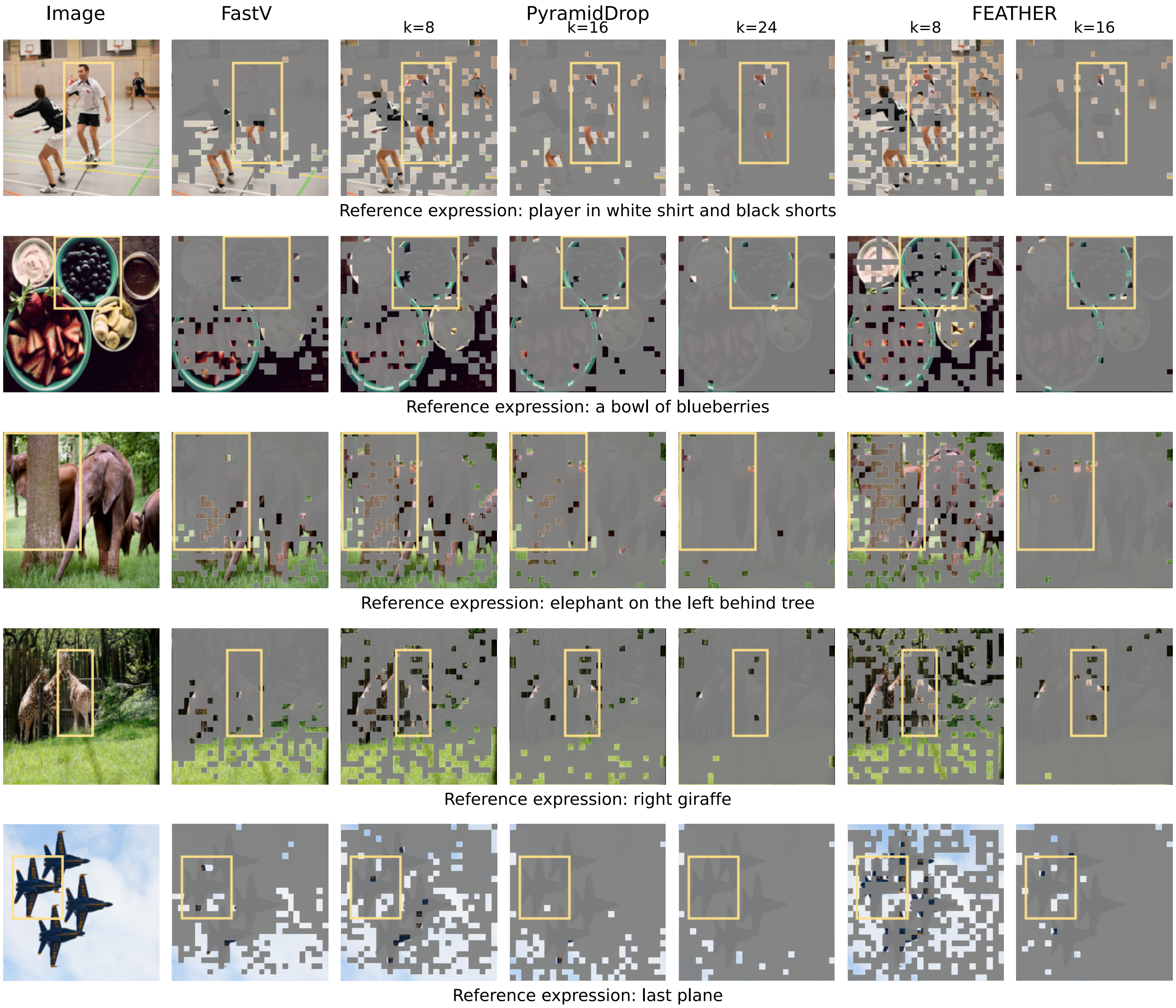


Image
FastV
PyramidDrop
k=8
k=16
k=24
FEATHER
k=8
k=16
Reference expression: player in white shirt and black shorts
Reference expression: a bowl of blueberries
Reference expression: elephant on the left behind tree
Reference expression: right giraffe
Reference expression: last plane

Figure A4. Visualizing the ability of FEATHER, FastV, and PyramidDrop to retain visual tokens relevant to the reference expression. We observe that our approach retains a substantially higher portion of tokens relevant to the reference expression.