Video, How Do Your Tokens Merge?

Supplementary Material

In the supplementary material we provide extra results for the speedup of different methods in Sec. 6.1, comparisons to other token reduction strategies for increasing/decreasing schedules in Sec. 6.2, and confusion matrices for ViViT in Sec. 6.4. We also provide information on training hyperparameters (for finetuning where it was required) in Sec. 7 and many more qualitative figures in Sec. 8 including extra merging examples, visualisations of layer decisions, and tests of semantic merging.

6. Additional Quantitative Results

In this section, we introduce extra results that we were not able to include in the main paper. We provide further scaling curves, results tables and confusion matrices on Kinetics-400 (K400) [16], Something-Something v2 (SSv2) [12] and EPIC-KITCHENS-100 (EK-100) [9].

6.1. Throughput Curve

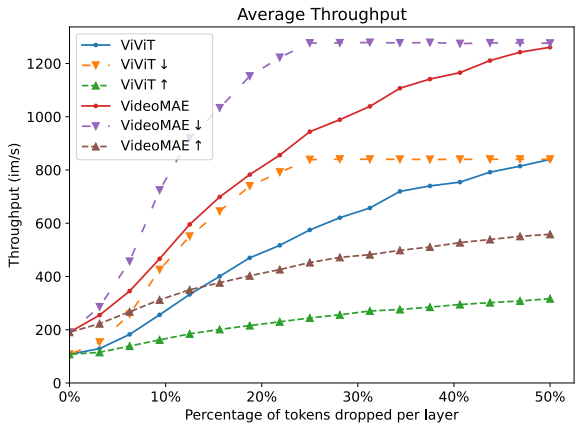


Figure 9. Curve corresponding to image throughput with ViViT and VideoMAE on K400 when increasing r (the number of tokens merged) up to its limit. The x-axis is the percentage (relative to the original total) of tokens dropped *per layer*.

With Fig. 9, we plot the throughput (in terms of images per second) of ViViT and VideoMAE on K400 for the constant, decreasing and increasing schedules. The increasing schedules introduce significantly slower speedups, showing that even with maximum merging per layer, it's only possible to introduce a speedup of roughly 2.5X. At the same proportion of merging, the decreasing and constant schedules meet at the same endpoints, achieving roughly 7X for VideoMAE and 8X for ViViT.

6.2. Analysis of Increasing/Decreasing Schedules

In Tab. 2, we have a table of results comparing reduction strategies for the decreasing schedule, using the same r value as in the main paper. Comparing across reduction strategies, we see that biasing merging towards the earlier layers is resulting in attention based dropout performing better than token merging for TimeSformer, Motionformer and VideoMAE. Random merging is essentially unuseable in this scenario, with accuracies approaching random performance in most metrics. Next, we directly compare the upper bound accuracy of the original models to those implementing a decreasing merging schedule. All models display large drops in accuracy, with VideoMAE in particular dropping by almost 50% on K400, 30% on SSv2 and 45% on EK-100. We've demonstrated that early transformer layers make merging decisions that do not correspond to visual segmentations (see Sec. 8 for more examples), suggesting that biasing merging towards earlier layers may introduce merging that quickly obfuscates visual features. ViViT retains more accuracy, especially on EK-100, where it drops by roughly 5% across the different label types. Looking to the speedup gained, we see that the throughput has been increased significantly to a 4X speedup, from the 2.5X speedup that the constant schedule demonstrates in the main paper. From this table, we can determine that a lower r value is required for the decreasing schedule, otherwise merging begins to be especially detrimental for models other than ViViT.

Table 3 conducts the same experiments for the increasing schedule. Interestingly, *across all models and datasets*, token merging is outperforming other reduction strategies and random merging remains the worst strategy. The action and noun accuracies on ViViT are actually *improved* a small amount by token merging, which suggests that merging tokens well in the later layers might refine video features slightly. Comparing the upper bound model accuracy with the merged counterparts, we see smaller drops in accuracy than in Tab. 2. For the divided space-time models, the accuracy on the temporally sensitive datasets (SSv2 and EK-100 verb accuracy) still demonstrate significant drops, showing that even when biasing merging towards the later layers, the models' ability to fuse temporal information is being hindered. ViViT and VideoMAE demonstrate drops in accuracy of *no more than* 2% *across all datasets*, while gaining a speedup of roughly 1.6X, indicating that an increasing schedule is an extremely "safe" option for these models, minimising accuracy trade-off. We have demon-

Model	r	Reduction	K400	SSv2	EK-100			FPS	Speedup (X)
					Action	Verb	Noun		
TimeSformer [2]	0	-	76.63	50.66	31.32	55.48	47.23	117.78	1.00
	18×8	random drop	28.38	9.80	1.38	18.68	6.12	361.01	3.07
		drop	**30.41**	**11.23**	**1.68**	19.45	**6.36**	359.59	3.05
		random merge	3.32	1.72	0.85	17.07	2.76	354.71	3.01
		merge	25.26	9.22	1.39	**19.90**	5.80	360.33	3.06
Motionformer [26]	0	-	70.50	61.39	35.02	61.09	46.72	99.79	1.00
	18×8	random drop	46.27	20.68	8.31	31.58	17.10	331.00	3.32
		drop	48.53	21.66	**10.14**	**33.70**	**19.47**	329.94	3.31
		random merge	17.12	6.24	1.58	22.23	5.35	328.19	3.29
		merge	**50.64**	**21.73**	8.91	33.18	18.05	334.57	3.35
VideoMAE [30]	0	-	62.09	64.58	35.70	61.49	46.89	186.72	1.00
	150	random drop	20.48	24.17	10.24	34.93	17.56	748.05	4.01
		drop	**23.08**	31.30	**11.99**	**37.91**	**19.86**	747.32	4.00
		random merge	1.04	2.99	0.57	14.60	2.11	735.87	3.94
		merge	20.82	**33.34**	10.88	36.31	18.06	742.91	3.98
ViViT [1]	0	-	63.43	50.63	35.82	58.19	51.59	106.00	1.00
	300	random drop	42.71	29.58	13.16	37.33	25.11	436.39	4.12
		drop	43.94	32.30	14.92	38.93	28.69	433.73	4.09
		random merge	2.67	2.03	0.73	17.64	4.14	432.76	4.08
		merge	**57.01**	**43.80**	**23.78**	**48.79**	**37.55**	439.26	4.14

Table 2. Performance of token merging with a decreasing schedule when compared to alternative methods of reducing token sequence length. Bold indicates the reduction methods that achieve highest accuracy on a given dataset. Grey rows correspond to the upper bound accuracy of the original model.

Model	r	Reduction	K400	SSv2	EK-100			FPS	Speedup (X)
					Action	Verb	Noun		
TimeSformer [2]	0	-	76.63	50.66	31.32	55.48	47.23	117.78	1.00
	18×8	random drop	72.36	22.03	19.36	40.29	37.73	163.89	1.39
		drop	72.72	27.54	21.98	43.78	39.88	166.17	1.41
		random merge	65.96	16.27	14.63	37.46	28.75	163.91	1.39
		merge	**74.24**	**28.91**	**23.28**	**45.23**	**41.56**	163.97	1.39
Motionformer [26]	0	-	70.50	61.39	35.02	61.09	46.72	99.79	1.00
	18×8	random drop	67.79	31.07	20.06	43.29	35.62	142.05	1.42
		drop	67.56	31.74	22.60	46.28	38.36	143.44	1.44
		random merge	64.86	30.06	18.04	41.54	32.37	142.56	1.43
		merge	**68.00**	**32.98**	**22.91**	**46.39**	**38.73**	141.77	1.42
VideoMAE [30]	0	-	62.09	64.58	35.70	61.49	46.89	186.72	1.00
	150	random drop	60.09	62.53	33.15	59.70	44.41	312.60	1.67
		drop	60.44	63.59	**34.35**	**60.56**	**45.85**	319.38	1.71
		random merge	48.32	55.00	22.91	50.56	33.15	316.12	1.69
		merge	**60.43**	**63.66**	34.34	60.28	45.24	311.66	1.67
ViViT [1]	0	-	63.43	50.63	35.82	58.19	51.59	106.00	1.00
	300	random drop	62.53	49.59	34.15	57.04	49.63	165.18	1.56
		drop	60.32	48.11	32.87	55.39	49.61	165.59	1.56
		random merge	51.94	37.41	21.25	6.33	36.07	164.64	1.55
		merge	**63.18**	**50.52**	**35.86**	**57.99**	**51.69**	164.13	1.55

Table 3. Performance of token merging with an increasing schedule when compared to alternative methods of reducing token sequence length. Bold indicates the reduction methods that achieve highest accuracy on a given dataset. Grey rows correspond to the upper bound accuracy of the original model.

Model	r	t	Reduction	K400	SSv2	EK-100		
						Action	Verb	Noun
VideoMAE [30]	0	-	-	62.09	64.58	35.70	61.49	46.89
	150	-	merge	56.10	**61.10**	31.27	**58.00**	42.39
		0.8	hybrid	**56.53**	61.04	**31.62**	57.90	**42.88**
ViViT [1]	0	-	-	63.43	50.63	35.82	58.19	51.59
	300	-	merge	63.08	50.15	35.11	57.24	**51.33**
		0.4	hybrid	**63.09**	**50.15**	**35.21**	**57.48**	51.30

Table 4. Performance of hybrid token merging with an increasing schedule when compared to vanilla token merging. Bold indicates the reduction methods that achieve highest accuracy on a given dataset. Grey rows correspond to the upper bound accuracy of the original model.

strated that with an increasing schedule and a reasonable r value, token merging is the reduction strategy that preserves accuracy the best.

6.3. Hybrid Merging

The final merged tokens are typically visually distinct clusters of image patches, a characteristic that can be observed in many qualitative examples in Sec. 8. One obvious limitation of the token merging [4] scheme is that it isn't adaptive in the sense that tokens are *forced* to merge if their pairwise similarity is one of the r largest. Towards the tail end of the merging process, different tokens become more visually dissimilar, as the clusters become saturated. We assume that merging these dissimilar tokens is destructive for performance, which we derive from the fact that random dropout is much preferable to random merging in Sec. 4.3.

To determine whether this phenomenon presents itself in vanilla token merging and attempt to alleviate it, we experiment with a hybrid scheme of dropout and merging. We a define a threshold t, where a token in the top r pairs is *dropped* instead of merged if the similarity is lower than t. Using this strategy, the model can adaptively merge/drop tokens, ensuring that merging only happens when token pairs are similar past a set threshold. In Tab. 4, we have the results of an experiment applying hybrid merging to ViViT and VideoMAE, after ablating the threshold against EK-100 to find an optimal value. Generally, these results highlight very similar performance to vanilla token merging, with the models showing consistent but small improvements on K400 and EK-100. Though marginal, these results indicate that it may be possible to develop stronger representations of token sequences with a combination of both dropout and merging.

6.4. ViViT Confusion Matrices

To investigate the errors introduced by merging with ViViT, we plot confusion matrices for the most frequent 10 verb and noun classes in EK-100. In Fig. 10 we have the difference in performance between a model *with* and a model *without* token merging. We can see a somewhat similar trend to the same figure produced for VideoMAE, with less predictions being introduced in the diagonal and more predictions elsewhere, though the model is clearly more resilient to confusion. For the verbs in Fig. 10 (left), we see that the model does not collapse towards the "take" and "put" classes. In fact, the largest change in performance sees "turn-off" being misclassified as "turn-on". These verbs are essentially the same action with slightly different context, suggesting that merging causes confusion among visually similar actions. Most of the nouns in Fig. 10 (right) do not shift significantly, however small objects like "spoon", "knife" and "sponge" are increasingly misclassified. As well as this, "sponge" is confused with "tap" particularly often. This again suggests that the features of small objects are the first to be lost by merging tokens.

7. Training Hyperparameters

As we've mentioned in the main paper, due to the fact that TimeSformer, VideoMAE and ViViT did not have freely available checkpoints for EK-100 online, we were required to finetune our own checkpoints for evaluation. In Tab. 5, we have an overview of the hyperparameters we used. These have been adapted (with as few changes as possible) from [1, 2, 30]. For TimeSformer, the learning rate is multiplied by 1, 0.1 and 0.01, at the first, twelfth and last epochs respectively. We do not use Mixup [36] when finetuning.

8. Qualitative Examples

Here we collect a range of qualitative examples that further our claims made in the main paper. To gather these, we generated visualisations at random and then kept a mixture of simple cases where (video subjects can be easily tracked across all frames) and more complex cases (where motion blur, occlusions or totally new subjects are introduced mid video).

First, we include a visualisation of tokens merging through a K400 clip in Fig. 11, where tokens of interest

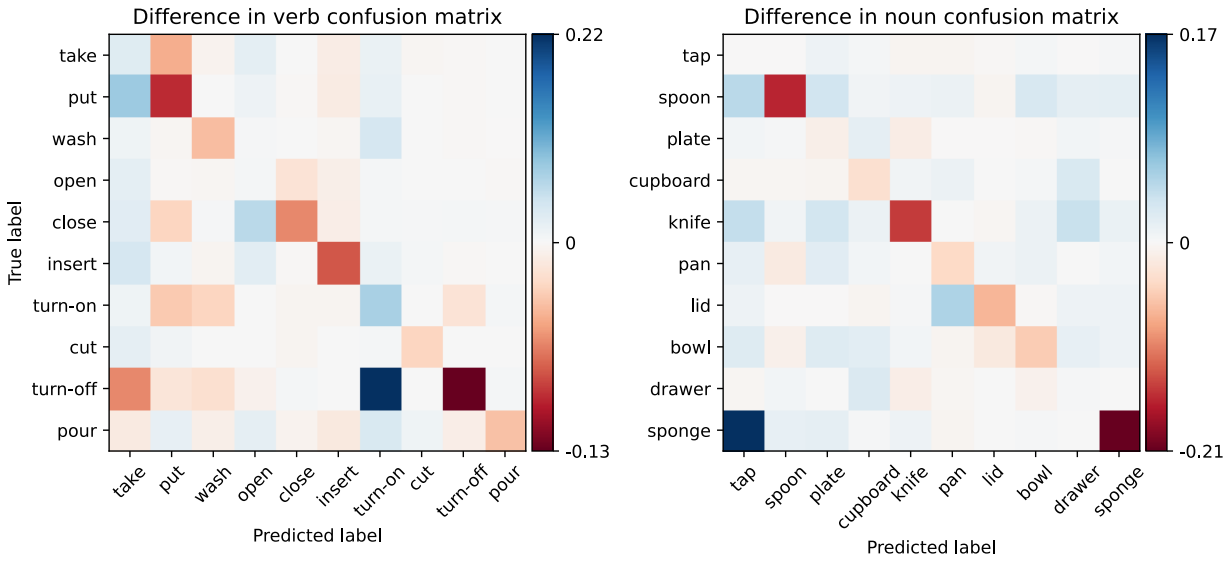


Figure 10. Impact on confusion matrices from Token Merging a ViViT model. The first 10 verb and noun classes are displayed left and right respectively from ViViT on EK-100. Red indicates less predictions and blue indicates more predictions.

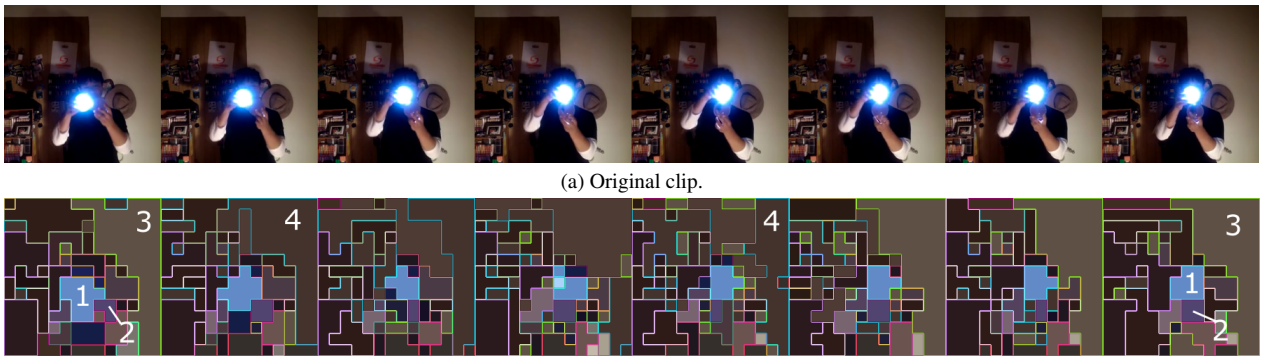
	TimeSformer	VideoMAE	ViViT
Batch size	128	128	64
Gradient accumulation steps	1	1	1
Base learning rate	5e-3	1e-3	1e-2
Learning rate policy	Step with relative LRs	Cosine with cosine warmup	Cosine with cosine warmup
Warmup learning rate	-	0	0
Warmup epochs	-	5.0	2.5
Epochs	15	50	50
Optimiser	SGD	AdamW	SGD
Momentum	0.9	0.9	0.9

Table 5. Hyperparameters used to train TimeSformer, VideoMAE and ViViT checkpoints with four H100 GPUs [24], used to evaluate merging on EK-100. Where possible we tried to reproduce the setup used in the original works.

have been numbered like in the main paper. In Fig. 11b, we have the final merged tokens for an example of "contact juggling". The bright blue ball is well captured across the entire clip by token 1, with the darker blue ball being captured by token 2. Notably, token 2 is tracked rotating clockwise around token 1. Tokens 3 and 4 represent the wall behind the right of the person, though the left side of the wall is not well merged, likely due to there being no common texture. In this case, video token merging is capable of tracking visually similar objects through all frames of the clip.

Next, we explore merging visualisations generated by both VideoMAE and ViViT. Firstly, from Fig. 12 to Fig. 16 we present visualisations of final merged tokens for K400, SSv2 and EK-100 respectively. Interestingly, in EK-100 examples exhibiting lots of head movement and motion blur, VideoMAE appears to handle these cases better by producing clearer segmentations of the clip. When hands are present near the centre of the frame, both models are capable of differentiating this from the objects the participant is interacting with.

Secondly, in Fig. 17 and Fig. 18 we collect more examples of the differences in merging outcomes for the first and last layers of the models. We note differences between how ViViT and VideoMAE merge tokens in the first layers, with VideoMAE creating large clusters with little consideration across frames, suggesting a spatial focus. Notably, the tokens tend to be divided into an upper and lower half, which could possibly be due to the distribution of foreground and background in EK-100, where the top of the frame will usually portray kitchen background and the bottom of the frame will typically contain interacting objects. On the other hand, ViViT tends to merge tokens across many frames within the first layer, yet these do not tend to occur around objects/distinguishable parts of the frame(s). Comparatively, the behaviour for the final layers (across both models) displays the merging of tokens around objects



(a) Original clip.

(b) Merged clip.

Figure 11. Visualisation of the final merged tokens for a K400 clip of "contact juggling", produced with VideoMAE. Token 1 tracks the bright blue ball, while token 2 tracks the darker ball rotating it.

in the scene and background.

Finally, in Fig. 19 and Fig. 20 we generate more examples of clips where frames from the most "similar" clip in EK-100 have been spliced in, to demonstrate the lack of semantic merging. In these cases, we have given the model the fairest chance by picking examples that also appear similar to the human eye. Much of the examples appear to demonstrate that the model is only merging within either the original clip *or* the spliced in clip, not between them. There appear to be some examples for which the participants' hands are merged between the spliced frames, likely due to the fact that there are few possible visual differences for these tokens. As discussed in the main paper, there is little to no evidence of the token merging process occurring between semantically relevant tokens, instead the token merging process is predominantly visual.

Figure 12. Visualisations of the final merged tokens for K400 clips, produced with VideoMAE.

Figure 13. Visualisations of the final merged tokens for K400 clips, produced with ViViT.

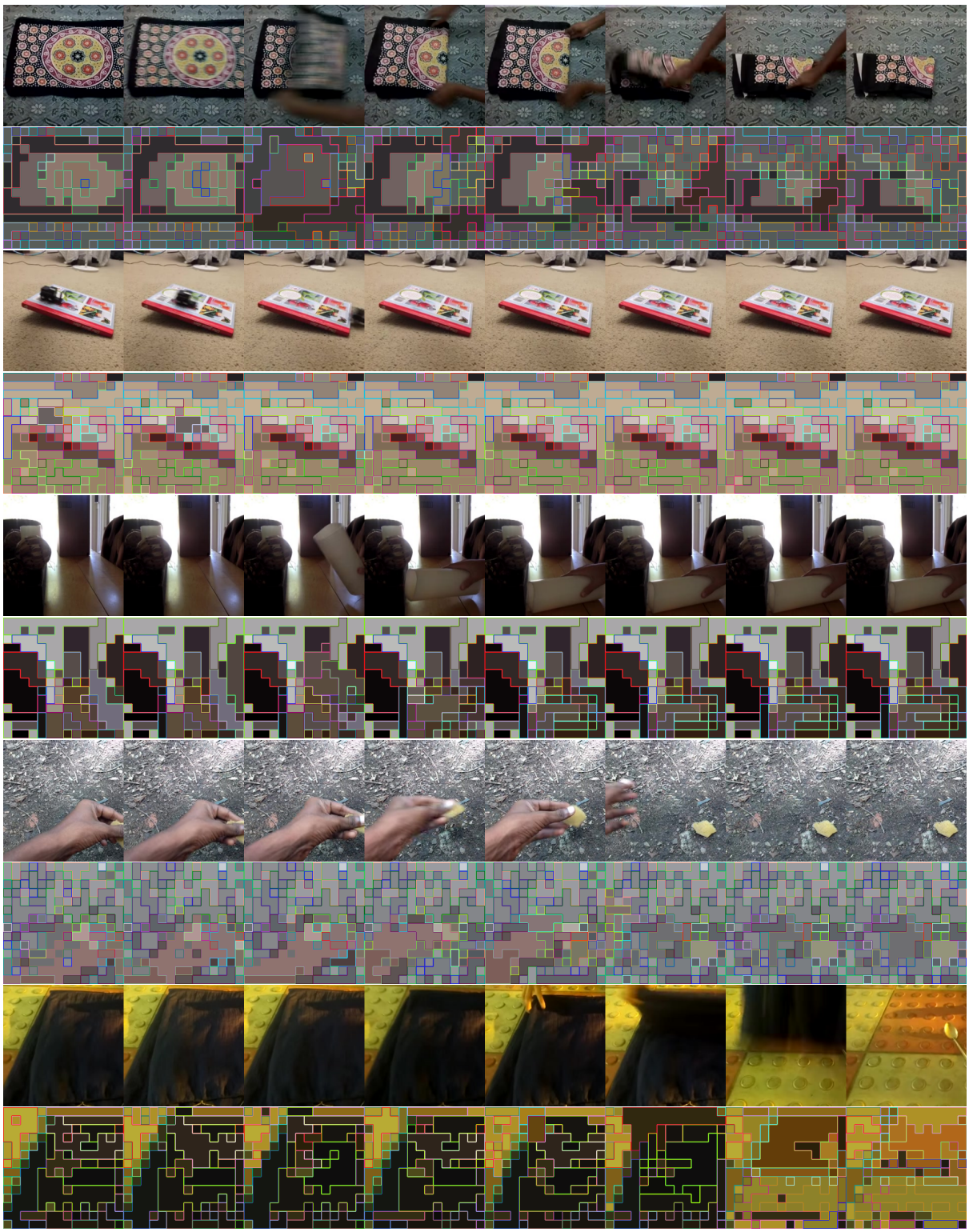


Figure 14. Visualisations of the final merged tokens for SSv2 clips, produced with ViViT.

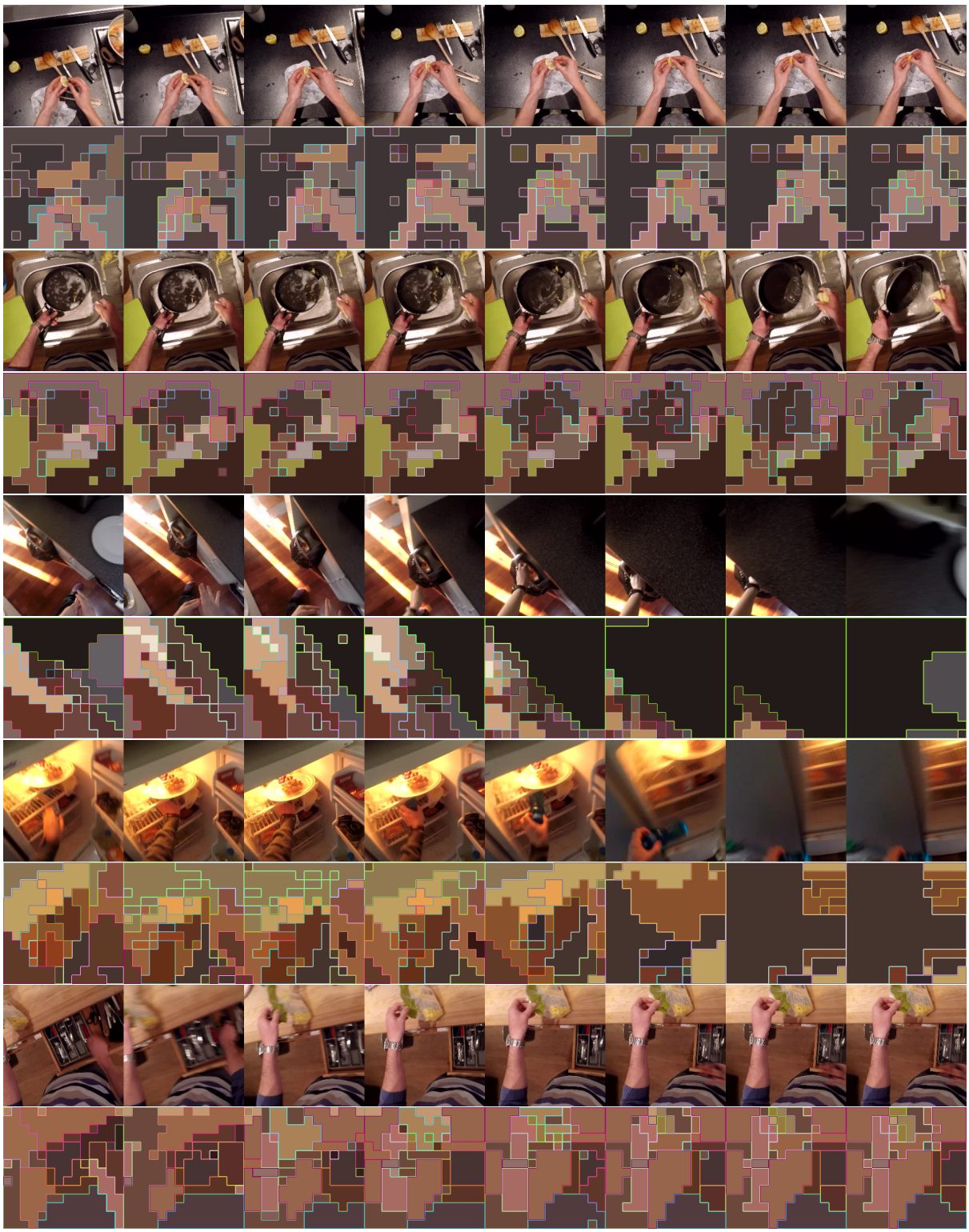


Figure 15. Visualisations of the final merged tokens for EK-100 clips, produced with VideoMAE.

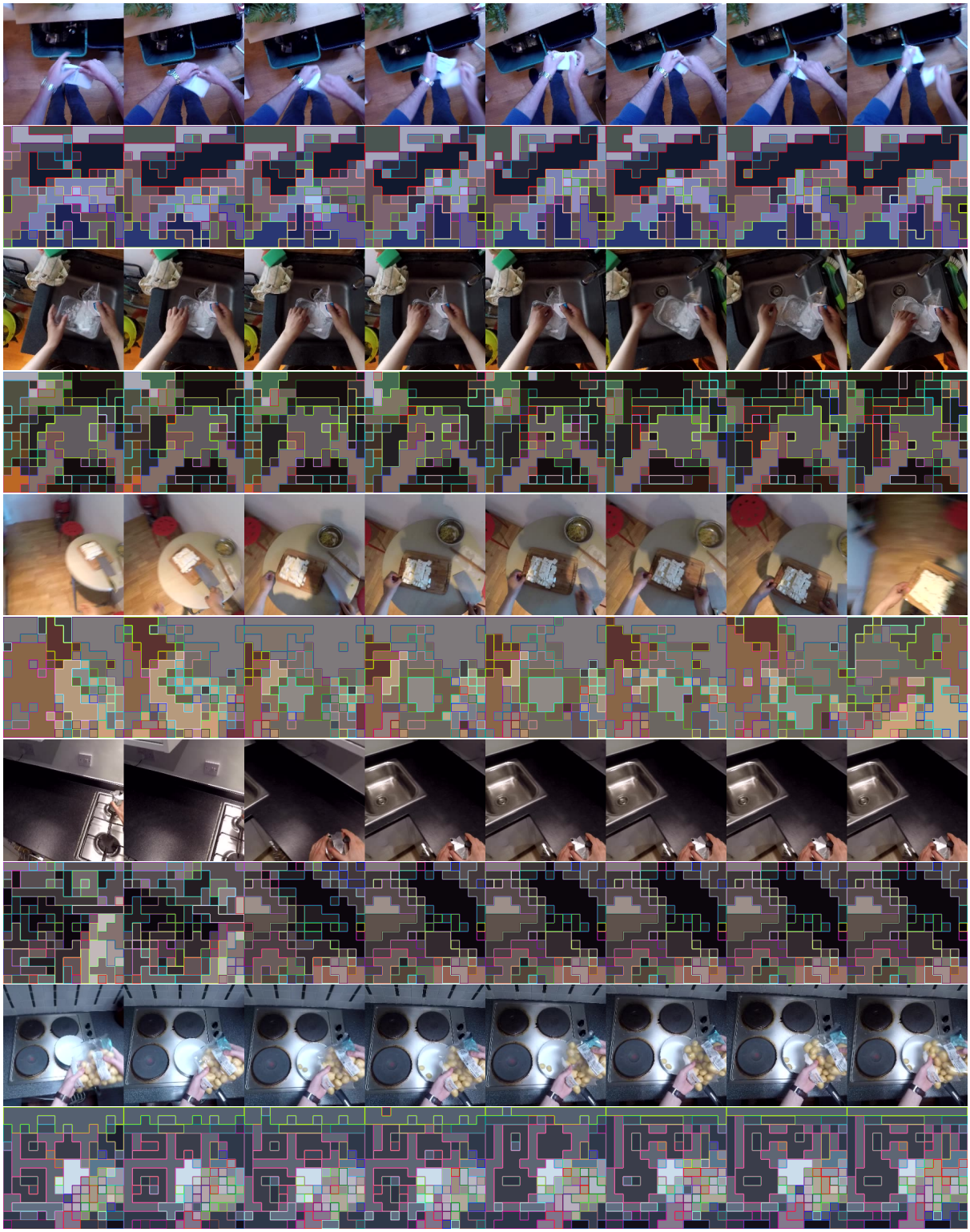


Figure 16. Visualisations of the final merged tokens for EK-100 clips, produced with ViViT.

Figure 17. Visualisations of the difference in merging decisions made in layer 1 versus layer 12, produced with VideoMAE.

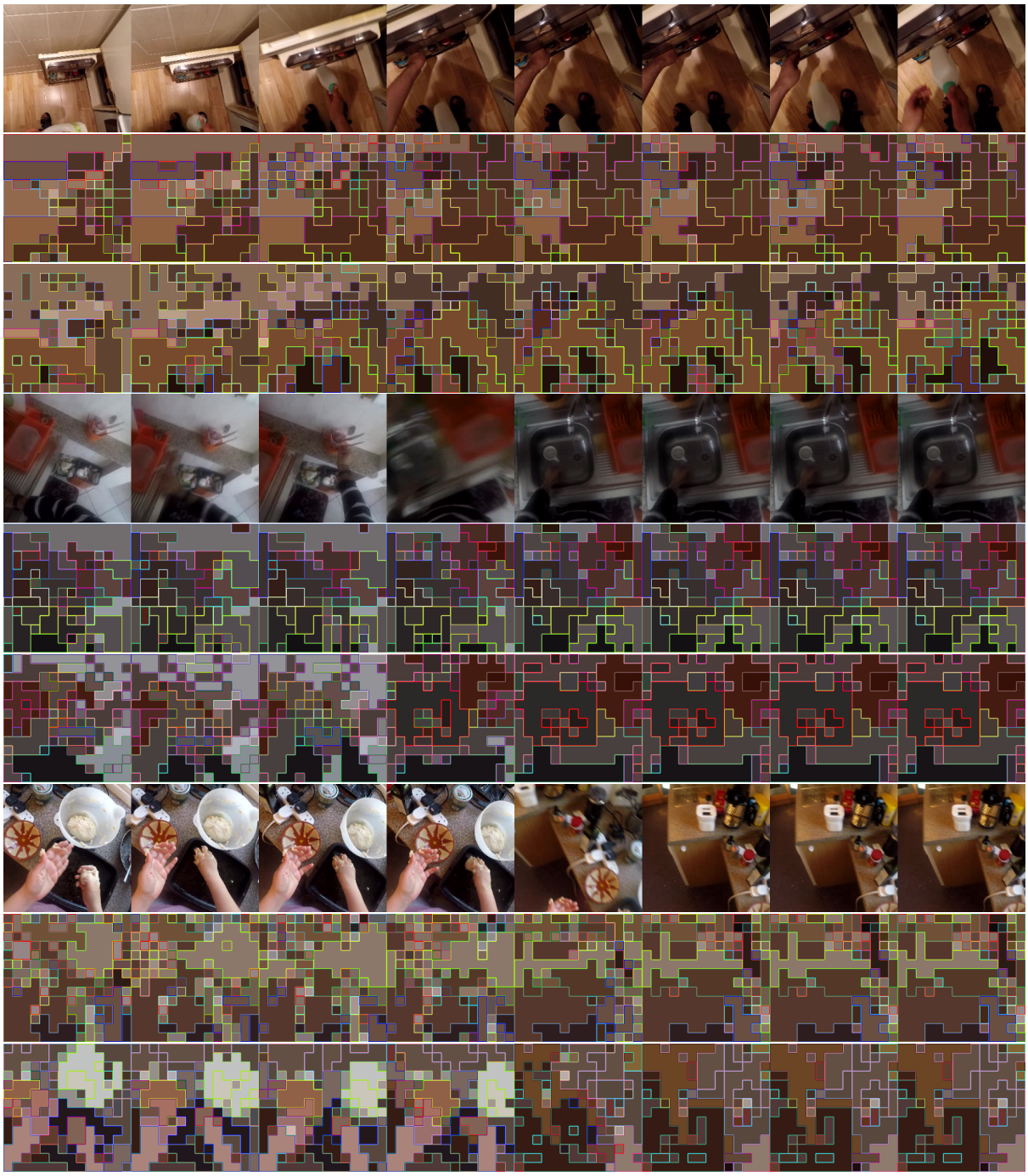


Figure 18. Visualisations of the difference in merging decisions made in layer 1 versus layer 12, produced with ViViT.

Figure 19. Merging outcomes for clips that have had half their frames from the most "similar" clip in the same noun class spliced in, produced with VideoMAE.

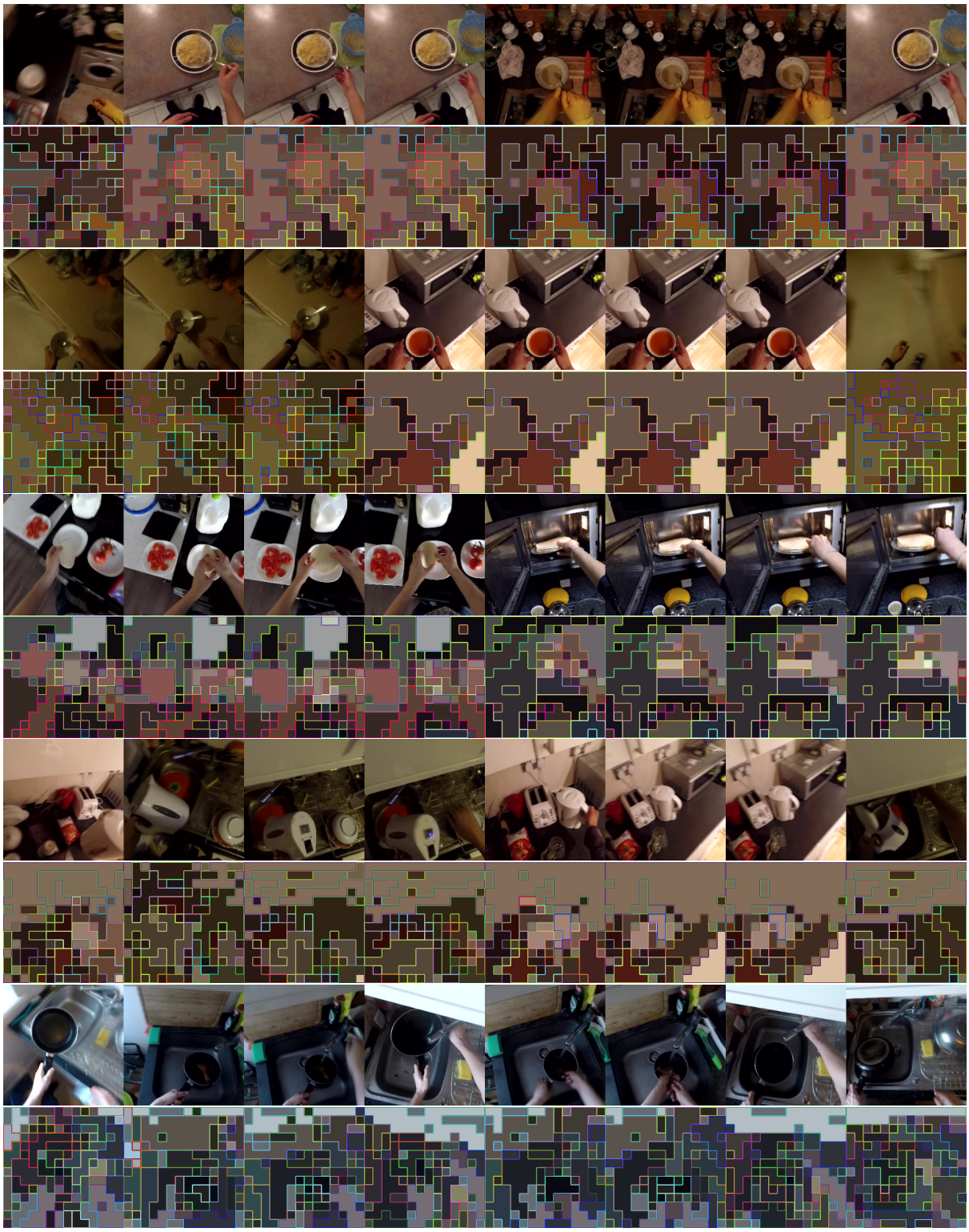


Figure 20. Merging outcomes for clips that have had half their frames from the most "similar" clip in the same noun class spliced in, produced with ViViT.