Leaping Into Memories: Space-Time Deep Feature Synthesis
Supplementary Material

S1. Additional Qualitative results

We demonstrated and overviewed qualitative results of inverted spatiotemporal models with LEAPS in Section 4.2. Supplementary to Figure 4, we provide additional results in order to visualize the features synthesized from different encoders when using the same action labels and stimuli. As shown in Figures S1 to S4 LEAPS can synthesize coherent visual features and effectively invert learned representations, independently of the spatiotemporal architecture used. Similar to the synthesized videos in Figure 4, for actions that are best described by the objects used e.g. *juggling balls*, *dribbling basketball*, and *playing trumpet*, all models optimize the input video to represent both class-relevant objects as well as actor-object interaction. Importantly, the synthesized videos show that video models learn motions with respect to both objects as well as actors. For the synthesized videos of *juggling balls* in Figure S1, the balls are primarily shown to be thrown upwards. In contrast, for the *dribbling basketball* videos in Figure S3, basketballs are bouncing on the side of the actor. In addition, evidence of LEAPS's ability to synthesize class-relevant features can be seen in Figure S4 where for the *playing trumpet* stimulus used, the better half of the trumpet is occluded. In actions that do not include or cannot be associated with specific objects; e.g. *baby crawling* in Figure S2, the synthesized videos primarily focus on the actor. This demonstrates that learned class-specific concepts of video models can be based on either objects, the actor's appearance and motions, or both, depending on the action performed.

Based on the videos from inverted models presented in Figures S1 to S4 there are no significant differences as to the objects and actors that are synthesized. However, the level of detail in the synthesized videos is shown to correlate with the model complexity. Specifically for *baby crawling* and *playing trumpet* synthesized videos from inverted models of increased capacities; e.g. X3D, Swin, and MViTv2 contain more visually distinct concepts than those of smaller architectures; e.g. 3D/(2+1)D Resnet-50. The effect is in line with the resulting synthesized videos from inverted models in Figure 6. Overall, LEAPS can invert models of varying complexities while also visualizing feature details based on the model's feature space capacity.

Model	λ_1	λ_L	r	$\mathcal{L}$
3D R50	1.0	0.3	$7.5e^{-3}$	7.892
(2+1)D R50	0.75	0.1	$5e^{-3}$	6.421
CSN R50	1.0	0.2	$5e^{-3}$	6.603
X3D$_{\text{XS}}$	1.0	0.2	$1e^{-3}$	5.175
X3D$_{\text{S}}$	1.0	0.1	$1e^{-3}$	5.538
X3D$_{\text{M}}$	0.75	0.1	$1e^{-3}$	6.387
X3D$_{\text{L}}$	0.75	0.1	$1e^{-3}$	7.190
TimeSformer	1.0	0.2	$2.5e^{-3}$	5.629
Video Swin-T	0.75	0.2	$1e^{-3}$	6.527
Video Swin-S	0.75	0.1	$1e^{-3}$	7.508
Video Swin-B	0.625	0.1	$1e^{-3}$	8.841
MViTv2-S	0.75	0.1	$2.5e^{-3}$	7.356
MViTv2-B	0.75	0.1	$1e^{-3}$	8.195
rev-MViT-B	0.625	0.1	$5e^{-3}$	7.227
UniFormerv2-B	1.0	0.2	$2.5e^{-3}$	6.053
UniFormerv2-L	1.0	0.1	$1e^{-3}$	7.415

Table S1: **LEAPS optimization hyperparameters** based on grid search. We additionally report the average loss on synthesized videos from the Kinetics validation set.

S2. Hyperparameter settings

As described in Section 4.1, we discover the optimal λ and r hyperparameters for each model through grid search. To limit the search space and computational overhead of hyperparameter tuning, we define $\lambda_1 \in \{0.5, 0.625, 0.75, 0.875, 1.0\}$, $\lambda_L \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, $r \in \{1e^{-3}, 2.5e^{-3}, 5e^{-3}, 7.5^{e-3}, 1e^{-2}\}$, where λ_1 is the priming weight for the first layer of the network, λ_L is the priming weight for the final layer of the network. Based on λ_1 and λ_L, we use a linear (decreasing) function for the remaining $\lambda \in \{2, ..., L-1\}$ layer priming weights. Table S1 provides a full list of the hyperparameters discovered and used for inverting each model. We note that the loss shows to increase in larger models due to the number of layers used for priming.

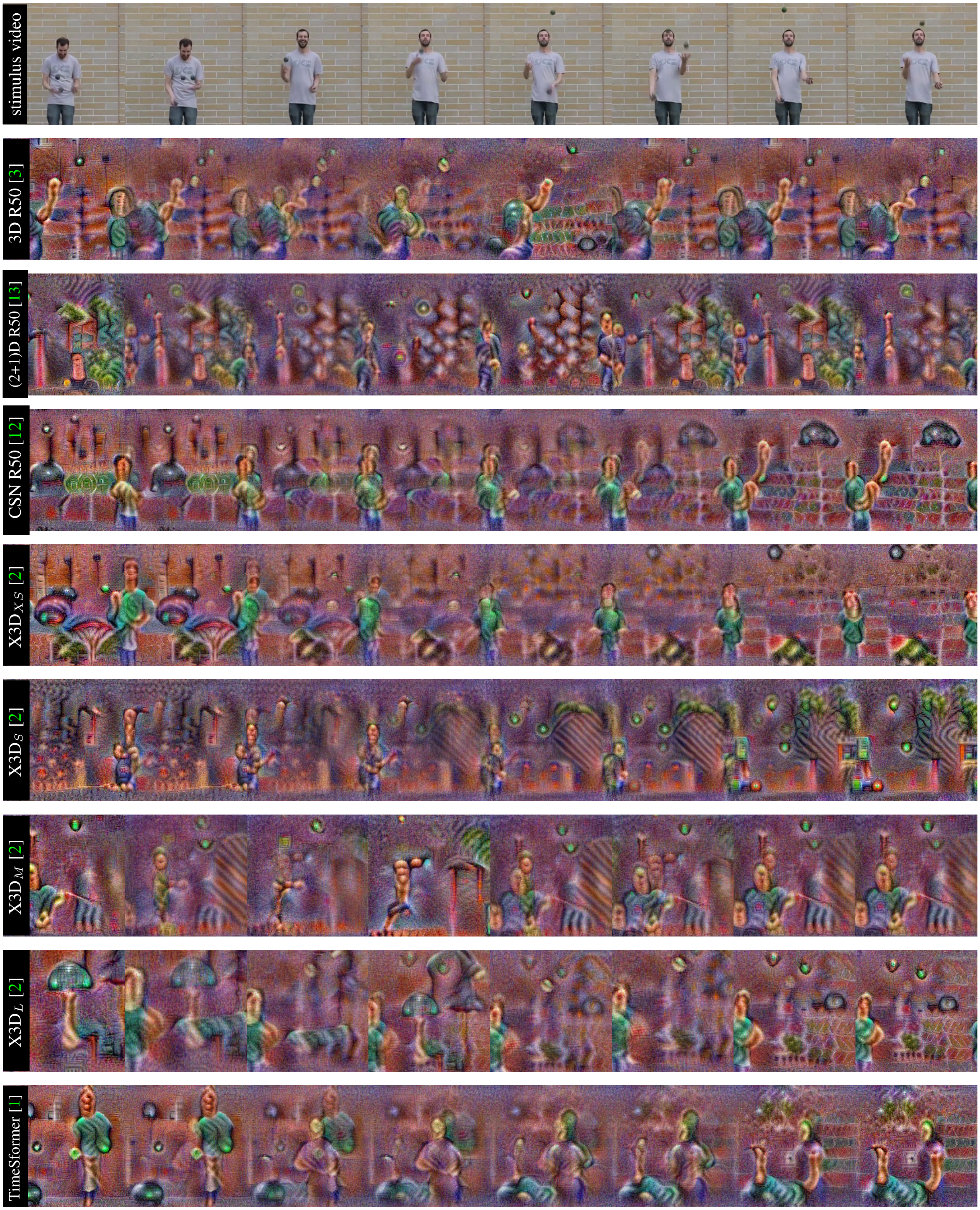


Figure S1: **Qualitative examples of synthesized features with LEAPS** for action label *juggling balls*.

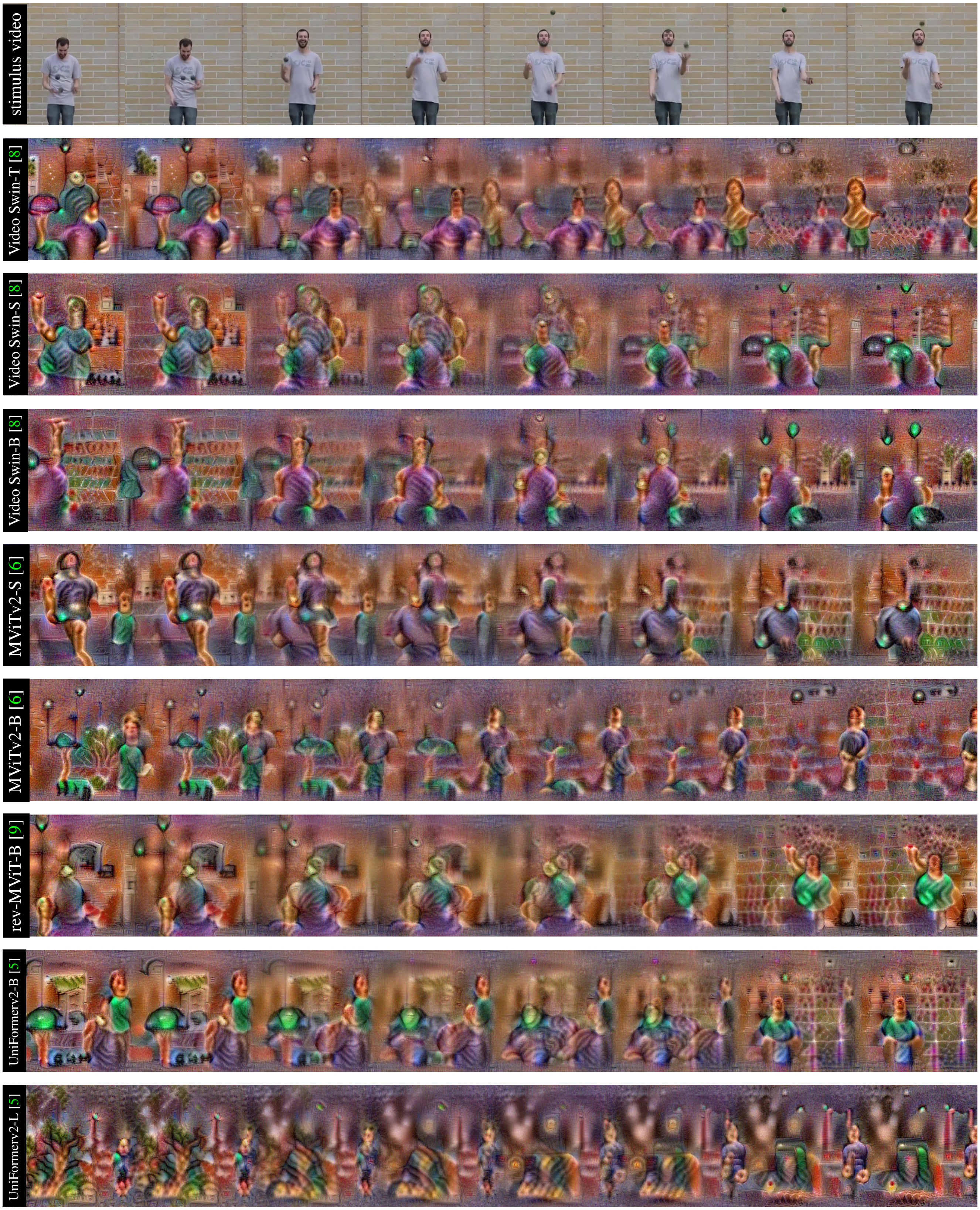


Figure S1: **Qualitative examples of synthesized features with LEAPS** for action label *juggling balls* (continued).

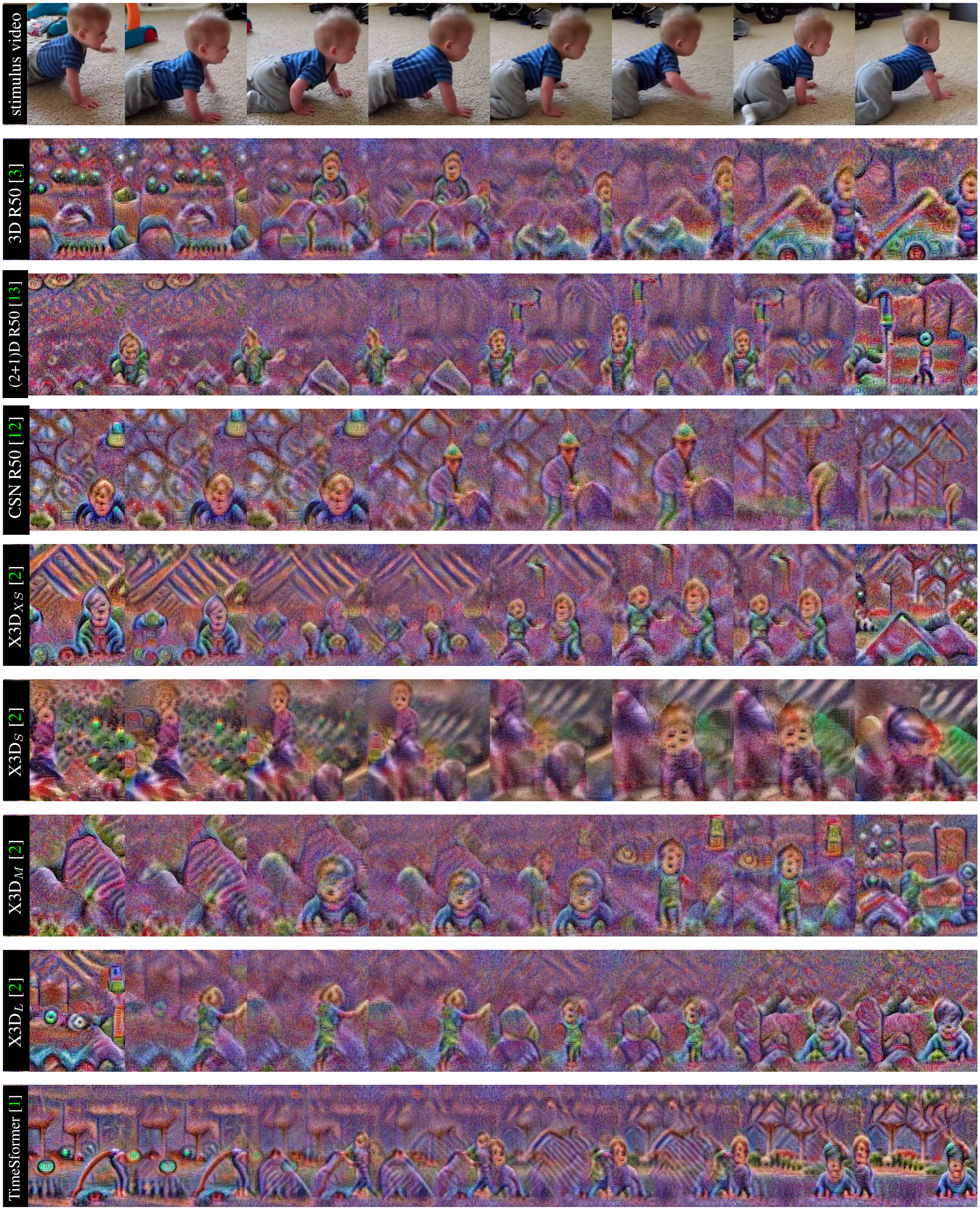


Figure S2: **Qualitative examples of synthesized features with LEAPS** for action label *baby crawling*.

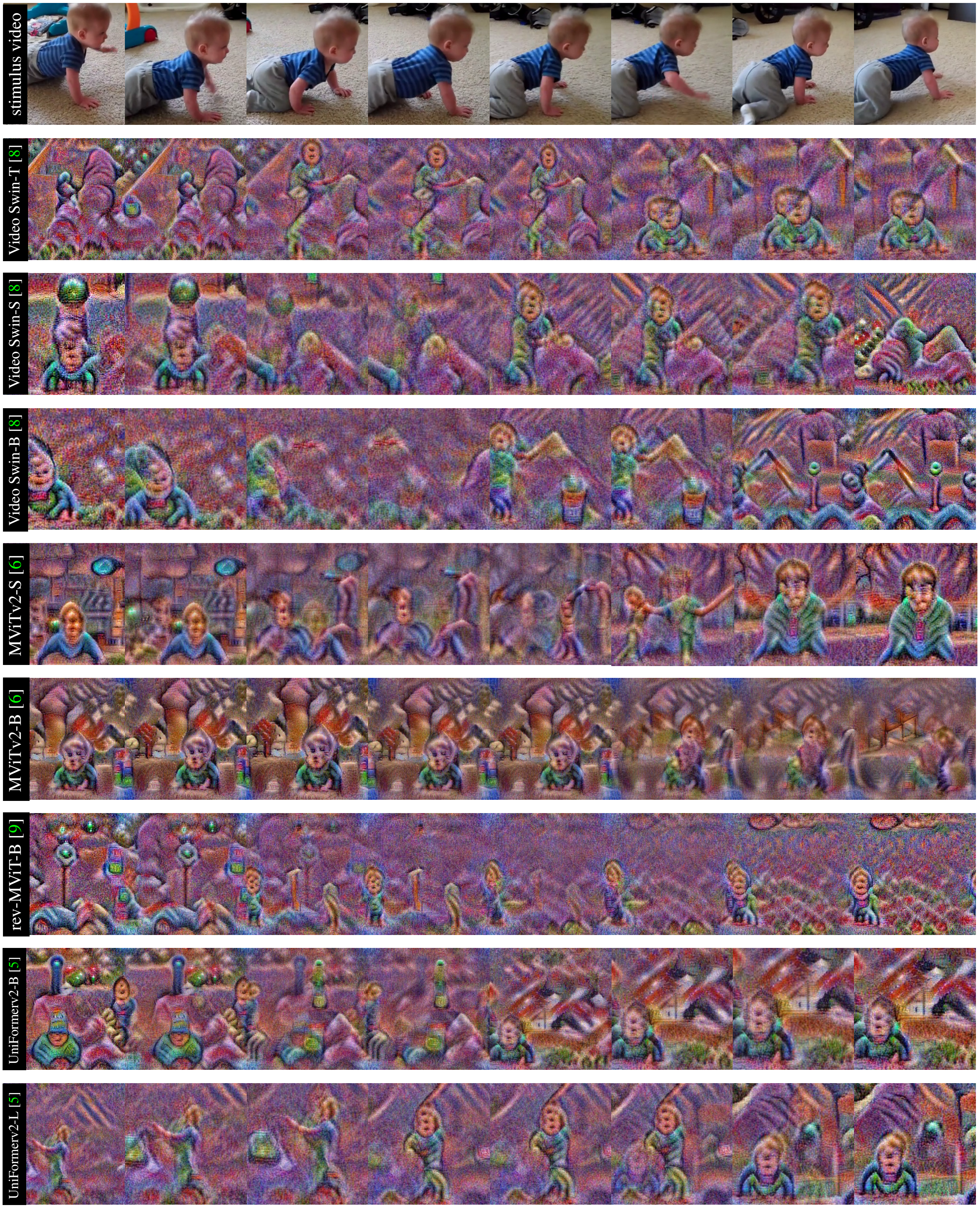


Figure S2: **Qualitative examples of synthesized features with LEAPS** for action label *baby crawling* (continued).

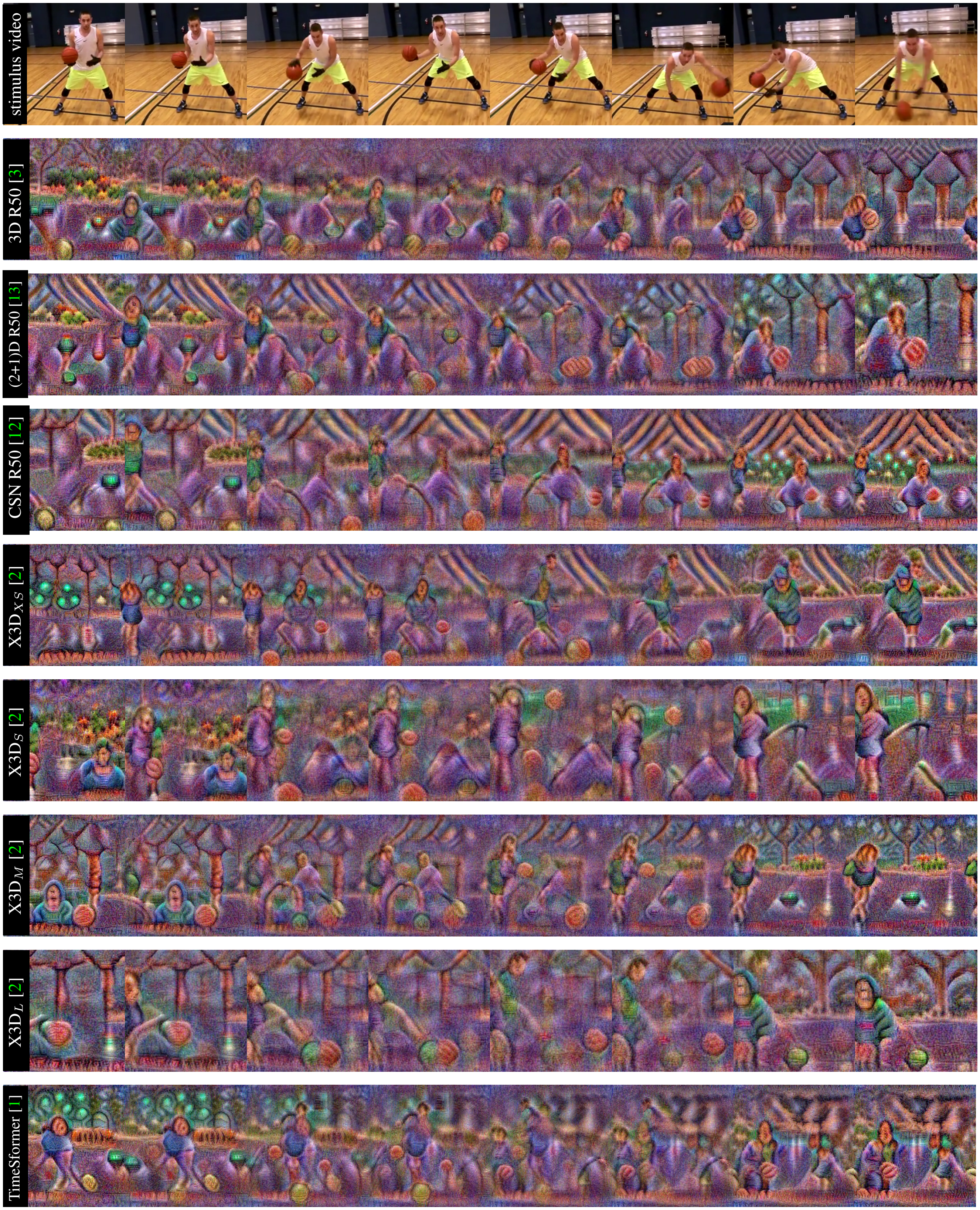


Figure S3: **Qualitative examples of synthesized features with LEAPS** for action label *dribbling basketball*.

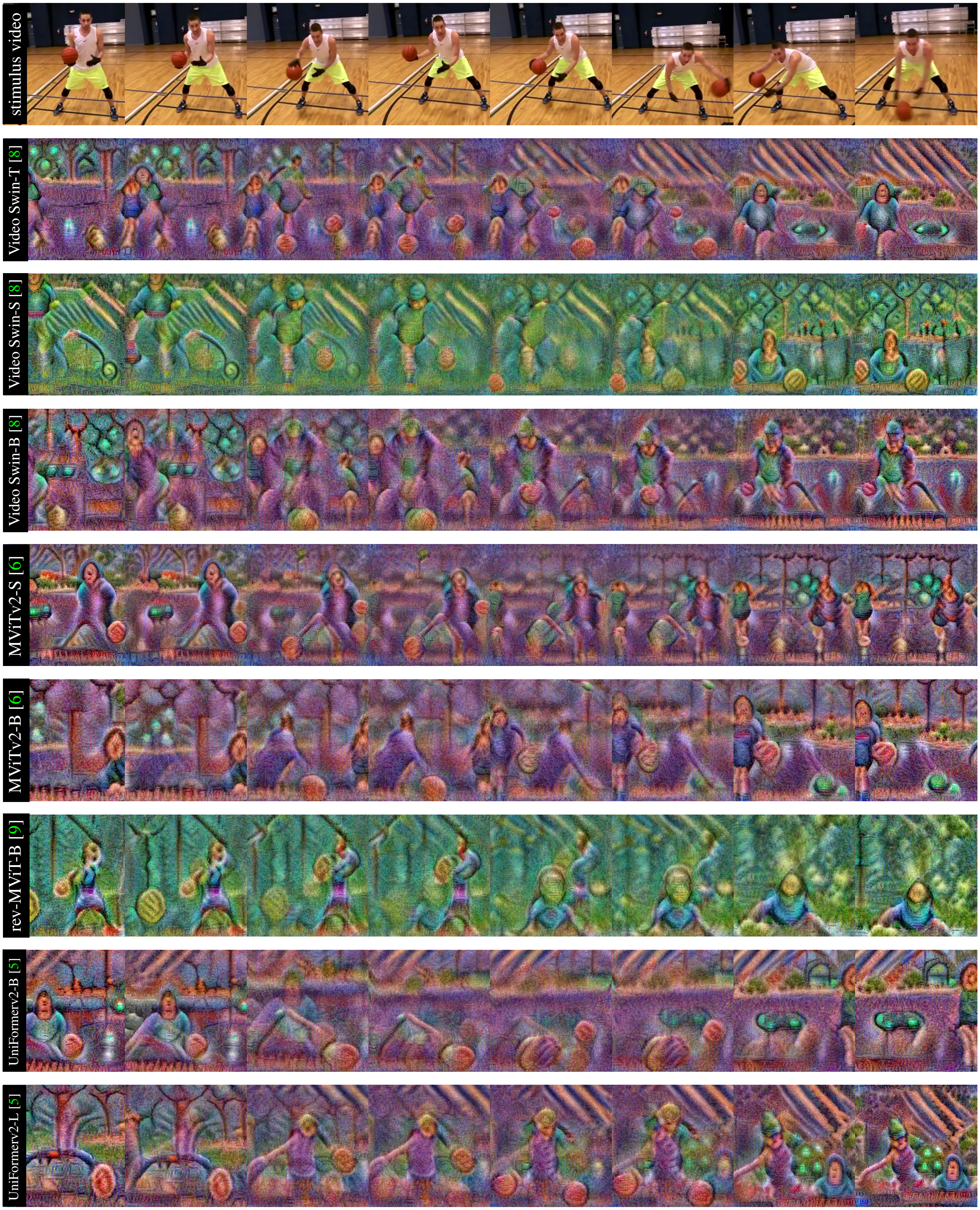


Figure S3: **Qualitative examples of synthesized features with LEAPS** for action label *dribbling basketball* (continued).

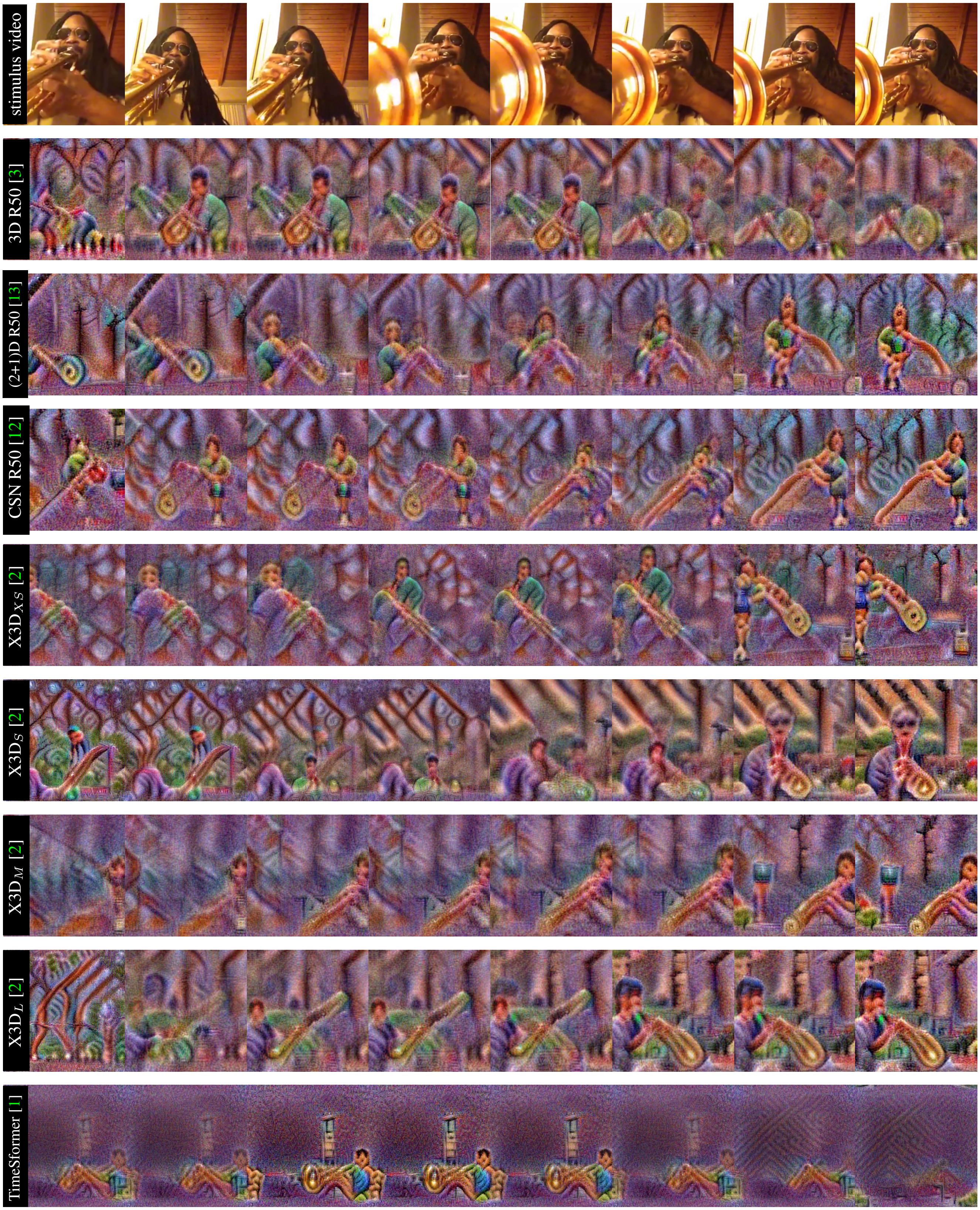


Figure S4: **Qualitative examples of synthesized features with LEAPS** for action label *playing trumpet*.

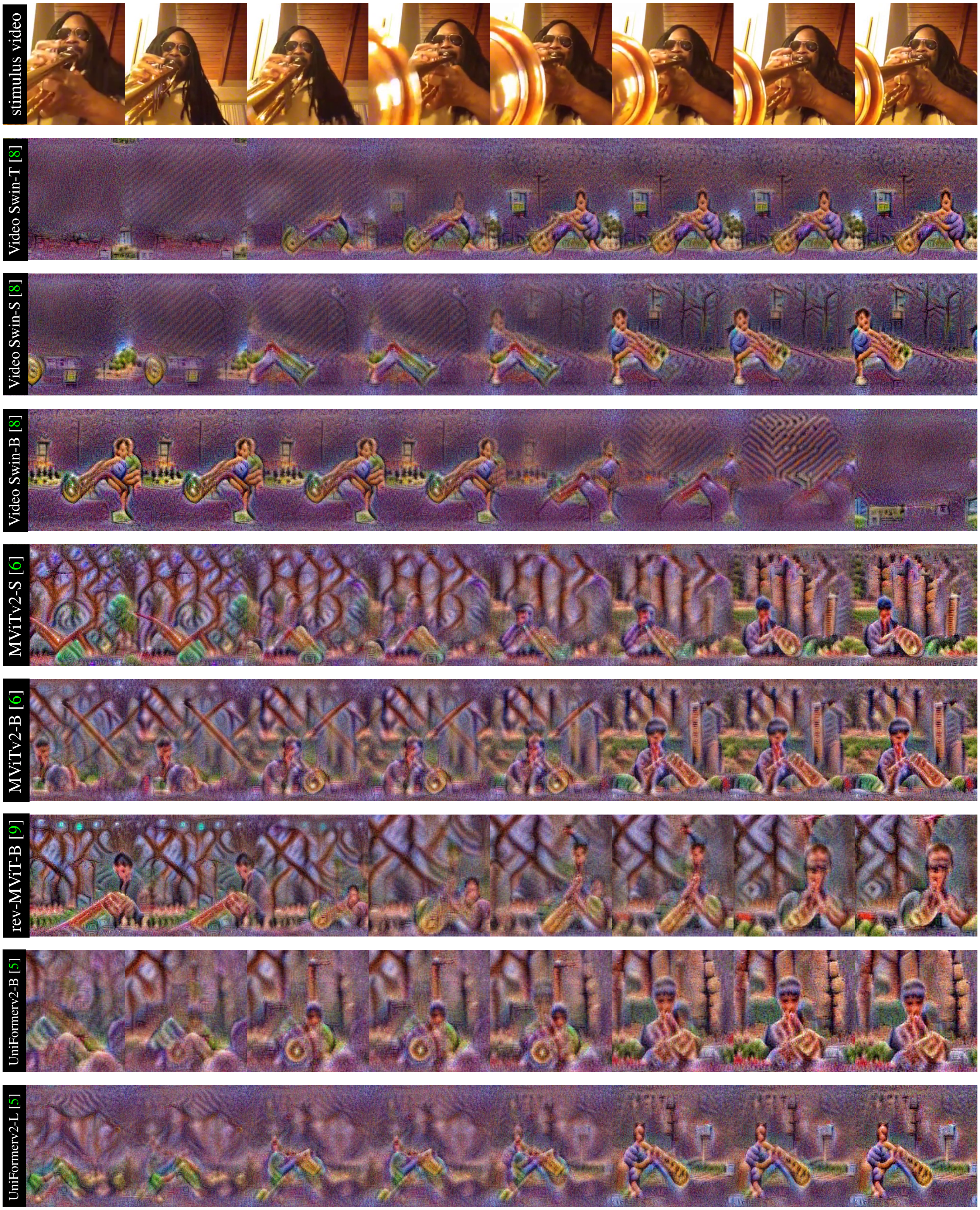


Figure S4: **Qualitative examples of synthesized features with LEAPS** for action label *playing trumpet* (continued).

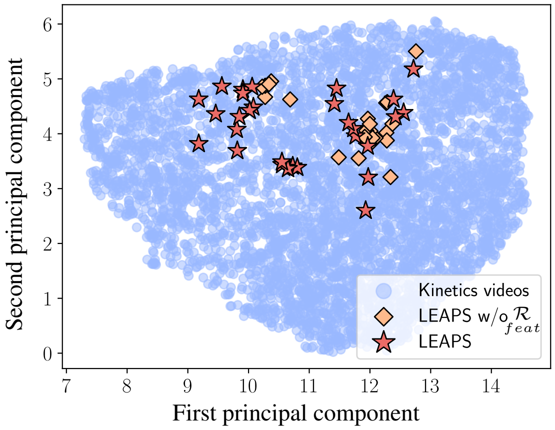


Figure S5: **Projection of X3D$_\mathrm{M}$'s final encoder layer embeddings** onto two principal components for Kinetic's class *tai chi*. Embeddings of videos from Kinetics are in blue, from LEAPS w/o $\mathcal{R}_{feat}$ in orange, and from LEAPS in red.

S3. Embedding space visualizations

LEAPS aims to synthesize visually coherent representations of inverted models. To better understand the relationship between the inverted model's features and real videos from the Kinetics-400 train set, we provide UMAP [10] visualizations of their feature embeddings for action *tai chi*. We use the spatiotemporally averaged feature vectors from the final convolution block in X3D$_\mathrm{M}$ (`s5.pathway0_res6.branch2.c`).

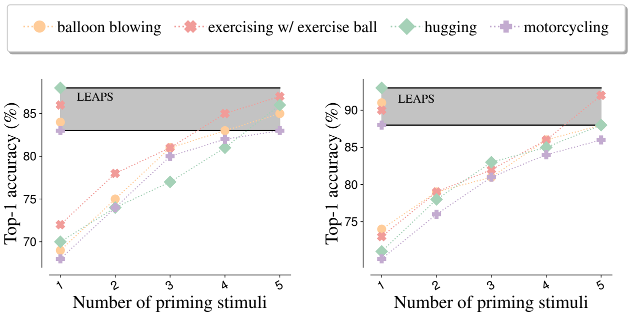
As illustrated from the results in Figure S5, inverted model embeddings are within the distribution of embeddings from Kinetics videos. While this is true for both embeddings from LEAPS synthesized videos as well as LEAPS synthesized videos without feature diversity regularization, LEAPS videos show a greater level of variation without being as closely concentrated as the embeddings of LEAPS w/o $\mathcal{R}_{feat}$.

S4. Priming layers

We further ablate over the number of layers used by the priming loss $\mathcal{L}_{prim}$. We select embeddings from the first 20%, 40%, 60%, and 80% of the total network layers for our priming loss. Given our proposed LEAPS uses embeddings from all network layers; i.e. $\mathbf{\Lambda} = \{1, ..., L\}$, each setting in turn uses $\mathbf{\Lambda}_{20} = \{1, ..., \lfloor \frac{L}{5} \rfloor\}$, $\mathbf{\Lambda}_{40} = \{1, ..., \lfloor \frac{2L}{5} \rfloor\}$, $\mathbf{\Lambda}_{60} = \{1, ..., \lfloor \frac{3L}{5} \rfloor\}$, and $\mathbf{\Lambda}_{80} = \{1, ..., \lfloor \frac{4L}{5} \rfloor\}$, where $\lfloor \cdot \rfloor$ denotes the floor function. As shown in Table S2, for both 3D R50 and X3D$_\mathrm{M}$, priming layer reductions also correspond to large decreases in top-1 accuracies and inception scores. The degradation in accuracy and IS is observed for both the inverted models as well as the verifier.

Priming	top-1 (%)		Inception Score (IS)	
layers (%)	model	ver.	model	verifier
3D R50				
20	19.0	4.1	1.3 ± 0.2	1.1 ± 0.1
40	23.4	9.5	1.8 ± 0.4	1.4 ± 0.4
60	41.8	23.4	2.5 ± 0.6	1.6 ± 0.5
80	69.3	54.6	4.2 ± 1.3	2.0 ± 0.4
100 (LEAPS)	**86.7**	**68.5**	**9.0 ± 1.0**	**5.7 ± 0.7**
X3D$_M$				
20	15.8	3.9	1.1 ± 0.1	1.0
40	18.3	5.4	1.4 ± 0.4	1.0
60	32.6	18.7	2.1 ± 0.8	1.2 ± 0.2
80	55.0	37.2	3.8 ± 0.7	2.1 ± 0.6
100 (LEAPS)	**90.3**	**82.5**	**11.4** ± **0.9**	**8.0** ± **1.4**

Table S2: **Ablation on the percentage of model's layers used for priming**. The best results per metric are in **bold**.



(a) 3D R50

(b) X3D$_\mathrm{M}$

Figure S6: **Top-1 inverted model accuracy (%) with priming** over stimuli videos. The area between the lower and upper class-accuracy bounds achieved by videos from LEAPS is shown in gray.

S5. Multi-stimuli priming

Our proposed video model inversion method is based on the approximation of embeddings that are relevant to specific actions. LEAPS uses the embeddings from a single priming example as stimulus. As an alternative, one may use additional stimuli videos to recall the learned preconscious of models associated with a class. We show in Figure S6 the top-1 accuracies achieved by 3D R50 and X3D$_\mathrm{M}$ when priming is performed with multiple stimuli instead of using LEAPS regularizers. As observed, the use of temporal coherence and feature diversity regularizers terms can perform favorably over internal representations from a small number of multiple stimuli. However, increasing the number of stimuli used show comparable performance to that achieved by LEAPS, thus advocating for an alternative to regularizers when access to more data is available.

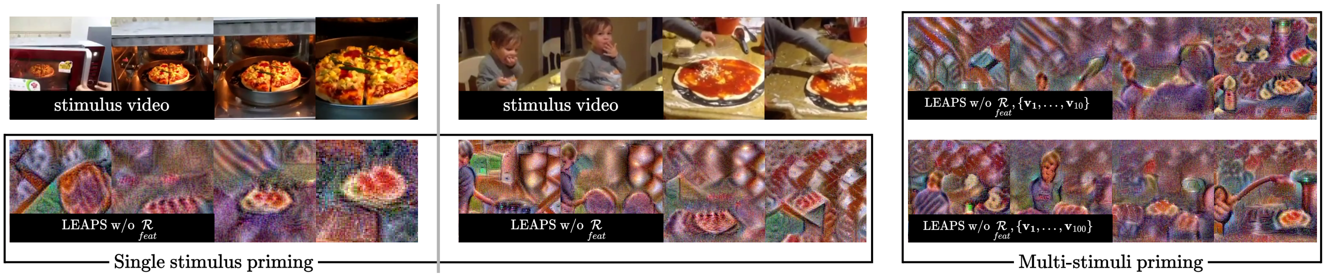

stimulus video
stimulus video
LEAPS w/o $\mathcal{R}_{feat}$ $\{\mathbf{v}_1,\ldots,\mathbf{v}_{10}\}$
LEAPS w/o $\mathcal{R}_{feat}$
LEAPS w/o $\mathcal{R}_{feat}$
LEAPS w/o $\mathcal{R}_{feat}$ $\{\mathbf{v}_1,\ldots,\mathbf{v}_{100}\}$
Single stimulus priming
Multi-stimuli priming

Figure S7: **Single and multi stimuli priming without $\mathcal{R}_{feat}$ for class** *making pizza*. The leftmost and center columns use a single but different stimulus video for model inversion. The right column uses the mean embeddings over 10 (top) and 100 (bottom) stimuli videos of the corresponding class. MViTv2-B features are inverted without a verifier network.

S6. Additional Discussions

Limitations. LEAPS is a general model-independent method for visualizing learned concepts of video models. We have demonstrated its effectiveness in inverting multiple architectures. As the synthesized visual features are not influenced by training data, with only a single stimulus video used to prime the network, we include a feature diversity regularizer. The regularizer uses the batch norm statistics as in [14], to approximate realistic features given a verifier network. The verifier is limited to architectures with batch norm layers and restricts the use of attention-based models.

We consider two approaches to mitigate this. The first approach is to remove the diversity regularized altogether. This evidently results in accuracy and IS decrease as shown in Table 3 with **LEAPS $\mathcal{L}_{prim} + \mathcal{R}_{coh}$** and **LEAPS (full)**. Qualitative examples are shown in the left and middle columns of Figure S7. The second approach is the use of multi-stimuli priming, which shows promise as an alternative in settings where additional data is available, as discussed in Section S5. We also provide examples of the effect of multi-stimuli priming at the rightmost column of Figure S7.

Applicability to other tasks. Our focus has been on the inversion of video models and the visualization of their embeddings. However, the method can be further extended to subsequent downstream tasks in the video domain including knowledge transfer [14], domain adaptation [7], counterfactual explanations [11] , and inversion attacks [4]. Such tasks have received little attention for video inputs and thus we believe that LEAPS can enable subsequent research efforts.

References

[1] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning (ICML)*, pages 813–824. PMLR, 2021. 2, 4, 6, 8

[2] Christoph Feichtenhofer. X3D: Expanding architectures for efficient video recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 203–213. IEEE, 2020. 2, 4, 6, 8

[3] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6546–6555. IEEE, 2018. 2, 4, 6, 8

[4] Ali Hatamizadeh, Hongxu Yin, Holger R Roth, Wenqi Li, Jan Kautz, Daguang Xu, and Pavlo Molchanov. Gradvit: Gradient inversion of vision transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10021–10030. IEEE, 2022. 11

[5] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer. *arXiv preprint arXiv:2211.09552*, 2022. 3, 5, 7, 9

[6] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4804–4814. IEEE, 2022. 3, 5, 7, 9

[7] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1215–1224. IEEE, 2021. 11

[8] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3202–3211. IEEE, 2022. 3, 5, 7, 9

[9] Karttikeya Mangalam, Haoqi Fan, Yanghao Li, Chao-Yuan Wu, Bo Xiong, Christoph Feichtenhofer, and Jitendra Malik. Reversible vision transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10830–10840. IEEE, 2022. 3, 5, 7, 9

[10] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. 10

[11] Jayaraman Thiagarajan, Vivek Sivaraman Narayanaswamy, Deepta Rajan, Jia Liang, Akshay Chaudhari, and Andreas Spanias. Designing counterfactual generators using deep

model inversion. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 16873–16884, 2021. 11

[12] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *International Conference on Computer Vision (ICCV)*, pages 5552–5561. IEEE, 2019. 2, 4, 6, 8

[13] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459. IEEE, 2018. 2, 4, 6, 8

[14] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8715–8724. IEEE, 2020. 11