Smooth Regularization for Efficient Video Recognition

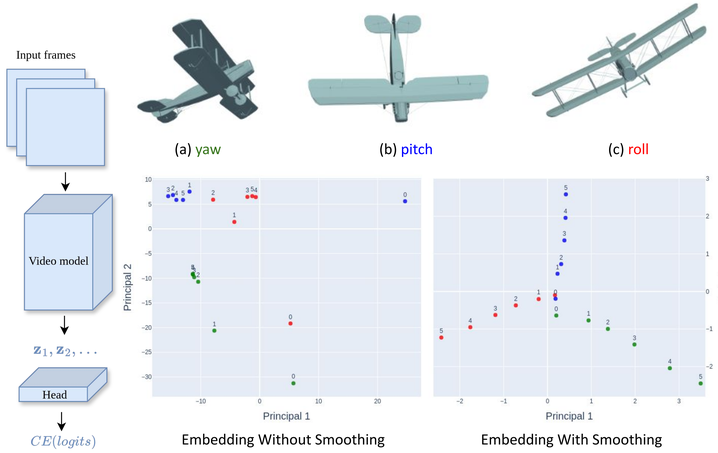
Gil Goldman* gilg@andrew.cmu.edu, Raja Giryes†, Mahadev Satyanarayanan*
* Carnegie Mellon University, Computer Science Department
† Tel Aviv University, School of Electrical and Computer Engineering
https://github.com/gilgoldm/grw-smoothing

NEURAL INFORMATION PROCESSING SYSTEMS

TL;DR

- GRW-smoothing enforces smooth intermediate embeddings in time.
- Plug-and-play loss for existing video models.
- +3.8–6.1 pp Top-1 accuracy on Kinetics-600 at similar FLOPs/memory.

Video Recognition Models are Not Smooth as a Function of Time



Warm-up Example. Top: Airplanes dataset with 1,000 training and 100 test videos of model airplanes rotating in yaw, pitch, or roll from random initial poses; single frames are ambiguous, so the task isolates temporal classification. Bottom: Embeddings of two identical models trained with and without GRW-smoothing; points are frame embeddings projected onto the first two principal components and colored by rotation class. The index is the frame index within the clip.

Setting

Consider a video frame sequence $X = (\mathbf{x}_t)_{t=0}^{M-1}$ and an intermediate layer output $\varphi(X) = Z = (\mathbf{z}_t)_{t=0}^{N-1}$ over time, where M and N denote the number of input frames and the number of embedding time steps (after any temporal subsampling), respectively. The goal of this work is to guide the optimization process to favor solutions φ for which $\mathbf{z}(t)$ is a piecewise smooth function of t.

The Gaussian Random Walk (GRW) Smoothing Term

Core Idea

Model the frame-to-frame embedding velocity as a Gaussian random walk.

Short Smoothing Time Windows

We divide $Z = (\mathbf{z}_t)_{t=0}^{N-1}$ into short subsequences:

$$Z^c = (\mathbf{z}_0^c, \ldots, \mathbf{z}_{T-1}^c) \coloneqq (\mathbf{z}_{cT}, \ldots, \mathbf{z}_{(c+1)T-1}),$$

where $c = 0, \ldots, C-1, \;\; C = \lfloor N/T \rfloor$.

Frame Ordering Contrastive Loss

$$\mathcal{L}_f(\varphi) = -\mathop{\mathbb{E}}_{X,c}\left[\log \frac{f(\mathbf{z}_0^c, \mathbf{z}_1^c, \mathbf{z}_2^c, ..., \mathbf{z}_{T-1}^c)}{\sum_{\pi \in S(1:T)} f(\mathbf{z}_0^c, \mathbf{z}_{\pi(1)}^c, \mathbf{z}_{\pi(2)}^c, ..., \mathbf{z}_{\pi(T-1)}^c)}\right],$$

where f is a smooth prior defined next, and S(1:T) is the group of all permutations π of the elements {1, . . . , T − 1}.

Smooth Prior

Define the velocities and accelerations of the embedding

$$\frac{d}{dt}Z^c = V^c = (\mathbf{v}_t^c)_{t=0}^{T-2} \coloneqq (\mathbf{z}_1^c - \mathbf{z}_0^c, ..., \mathbf{z}_{T-1}^c - \mathbf{z}_{T-2}^c),$$
$$\frac{d}{dt}V^c = A^c = (\mathbf{a}_t^c)_{t=0}^{T-3} \coloneqq (\mathbf{v}_1^c - \mathbf{v}_0^c, ..., \mathbf{v}_{T-2}^c - \mathbf{v}_{T-3}^c).$$

To smooth $\mathbf{z}(t)$, we model the distribution of the velocities as a random walk with Gaussian increments,

$$\mathbf{v}_t^c | \mathbf{v}_0^c = \mathbf{v}_0^c + \sum_{i=0}^{t-1} \mathbf{a}_i^c, \qquad t = 1, \ldots, T-2,$$

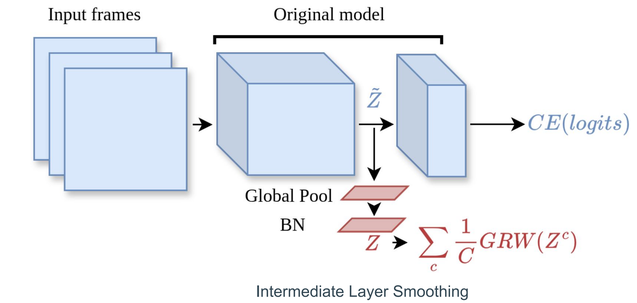
where $(\mathbf{a}_t^c)_{t=0}^{T-3}$ are i.i.d. $\mathbf{a}_t^c \sim \mathcal{N}(\mathbf{0}, I)$. Under this assumption

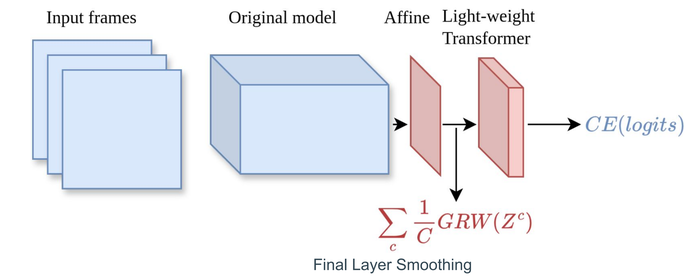
$$f(Z^c) \coloneqq p(\mathbf{v}_1^c, \ldots, \mathbf{v}_{T-2}^c | \mathbf{v}_0^c) = p(A^c) = \prod_{t=0}^{T-3} \mathcal{N}(\mathbf{a}_t^c),$$

where $\mathcal{N}$ denotes the density of the standard normal distribution.

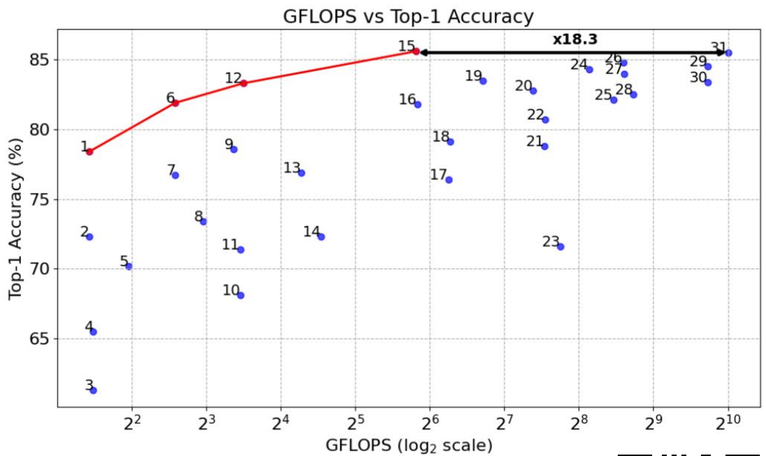
Applying GRW-smoothing to a neural network

GRW-smoothing can be plugged into intermediate or final layers with negligible extra cost.





Experiments



Results on Kinetics-600. Accuracy vs. FLOPs. Blue points correspond to all published models. GRW-smoothing improves the state-of-the-art performance of efficient models by 3.8–6.1 pp.