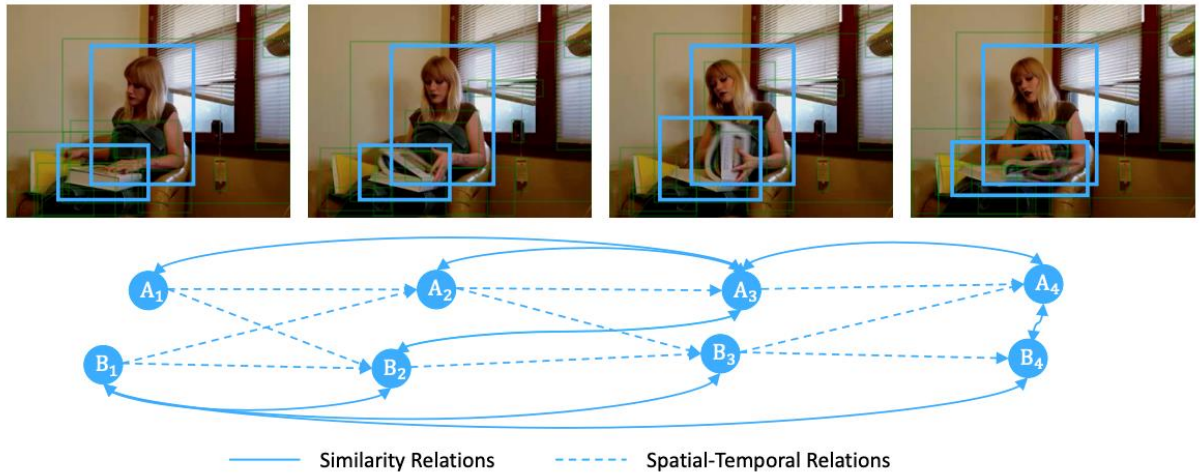
Videos as Space-Time Region Graphs

Jaedong Hwang

Computer Vision Lab.

Seoul National University



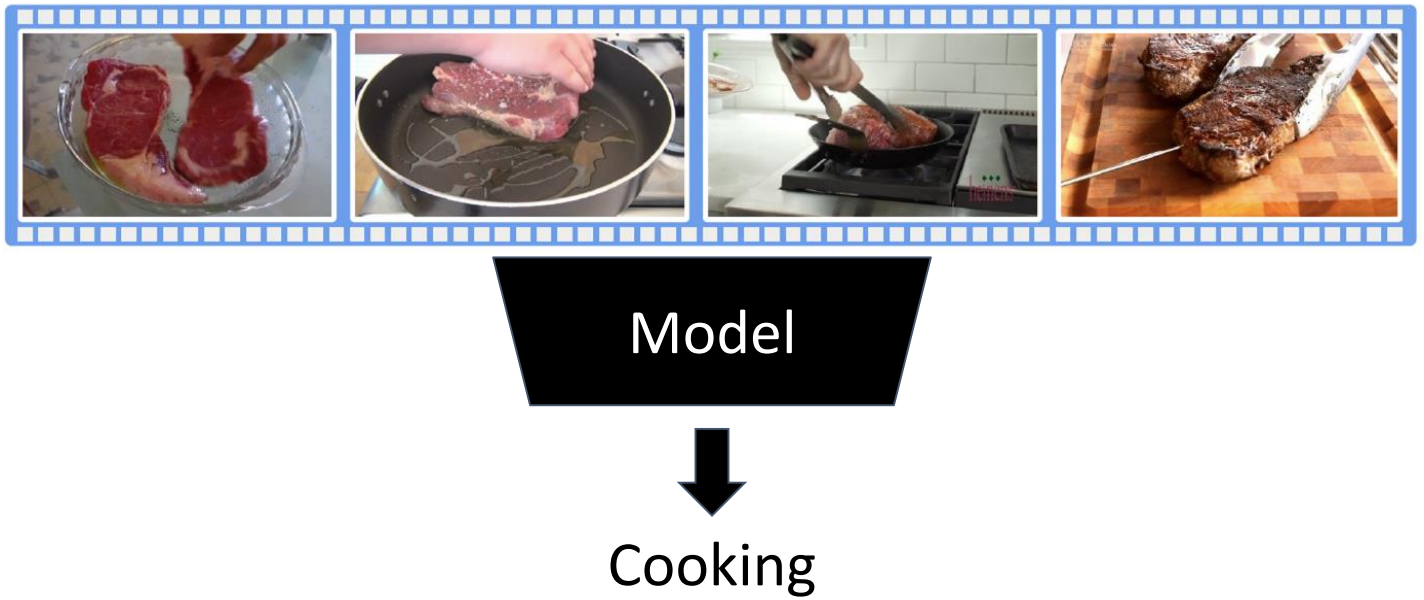
- TL;DR: Generate and utilize graphs of region proposals to capture relationship between objects.

- Codes are available on https://github.com/jd730/STRG

Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

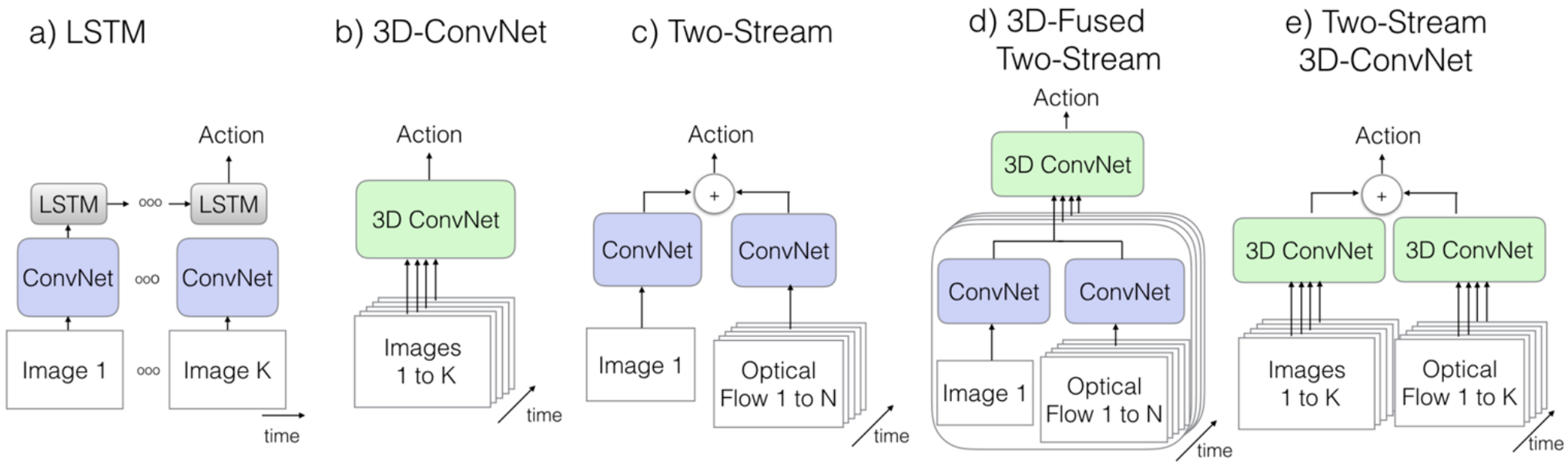
What is Action Recognition?

- Classify which actions in a videos.



- Depending on temporal direction, label can be changed
  - Door open vs. Door close

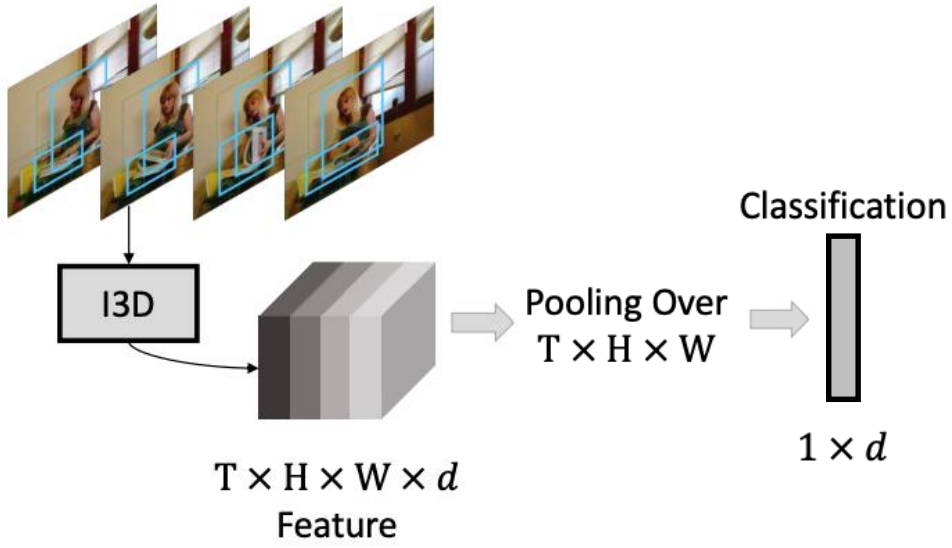
Various Architectures for Action Recognition



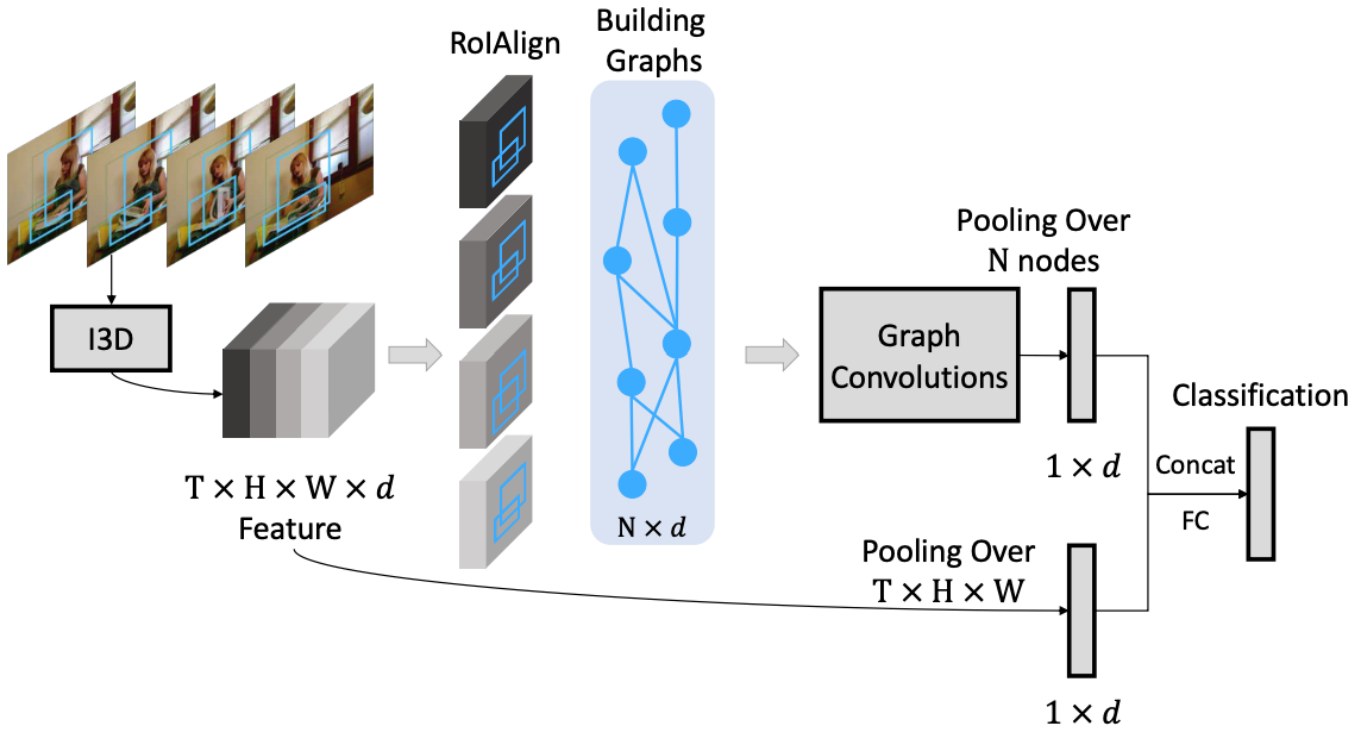
- Using 3D Convolution improves the performance by using temporal information.
- However, it is heavy and requires lots of computation.

Carreira, Joao, and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset." *CVPR*. 2017.

Videos as Space-Time Region Graphs



Previous architecture



Proposed architecture

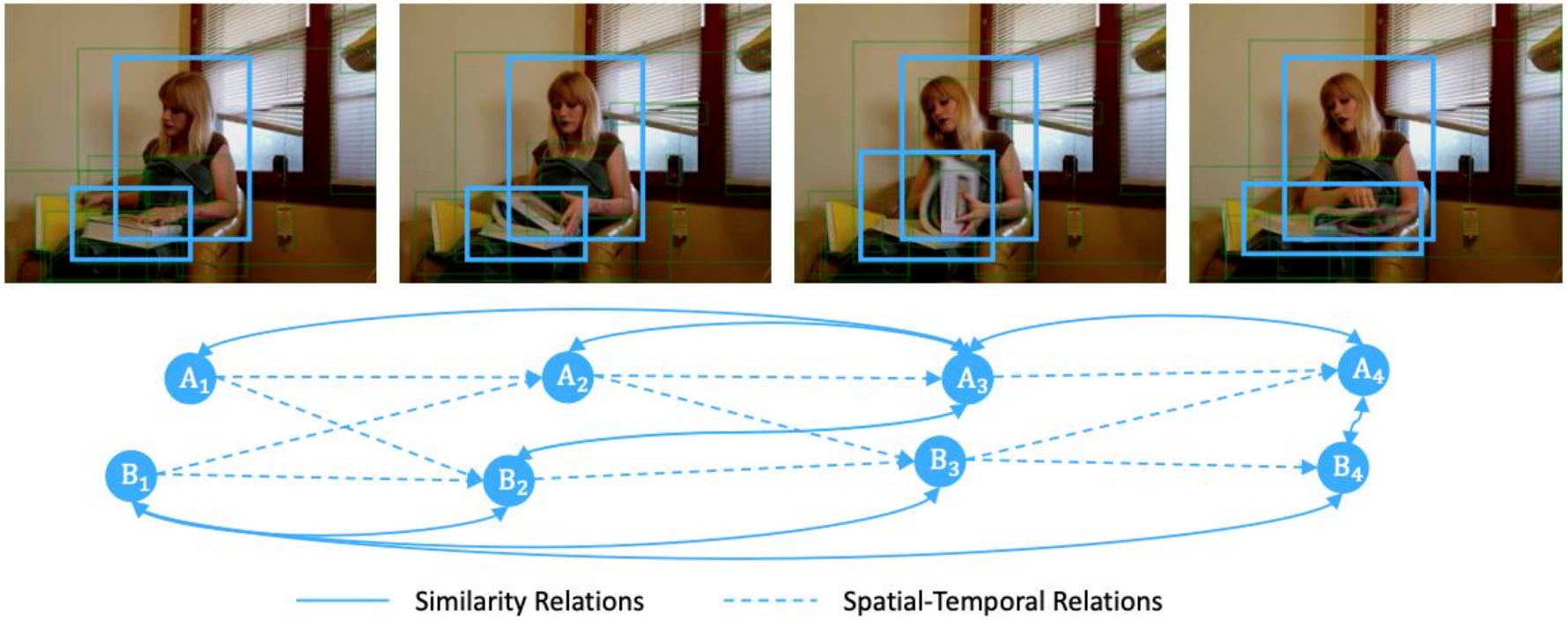
Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Videos as Space-Time Region Graphs

- Extract region proposals from RPN
- Extract feature for each proposal using RoI Align
- Generate graphs; spatiotemporal graph and similarity graph.
- Graph Convolutional Network

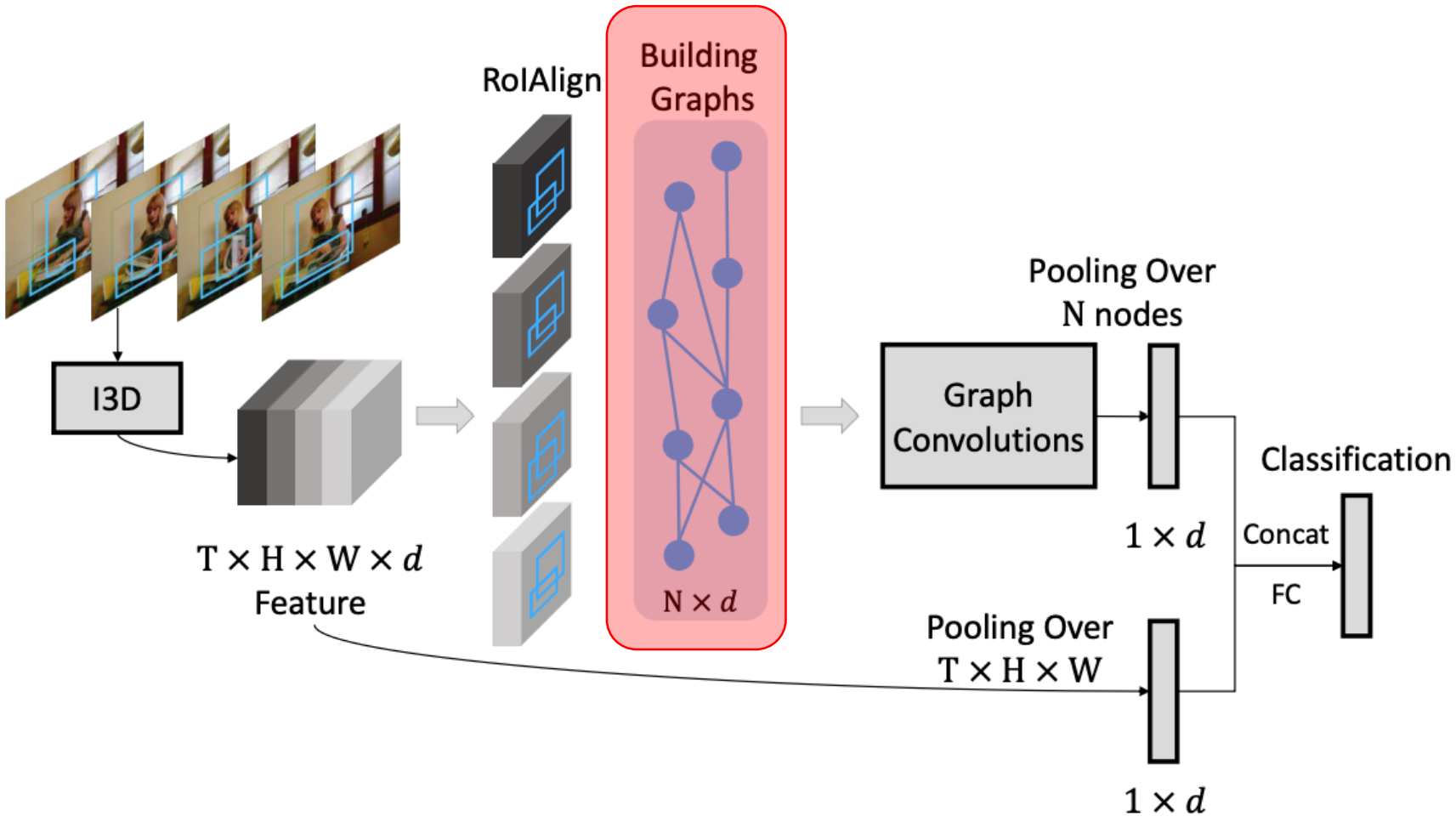
Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Two Graphs for region proposals



- Similarity graph – fully connected, similarity
- Spatiotemporal graph – partially connected, overlap

Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.



Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Building Graphs – Similarity Graph

- Fully connected directed graph.
- Transition matrix, $\mathbf{G}_{ij}^{sim}$ is defined as correlation btw embedded features.

$$F(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi'(\mathbf{x}_j)$$

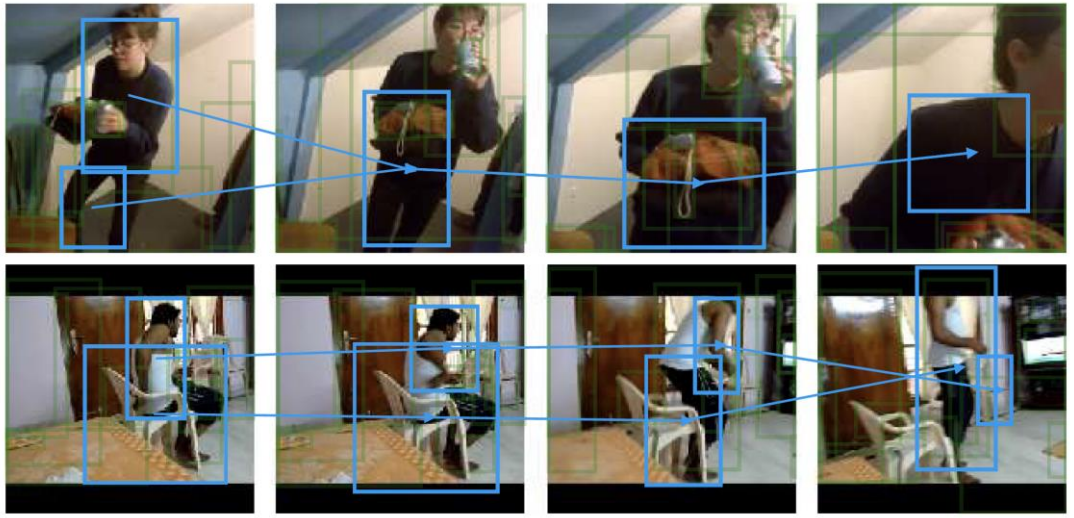
where $\phi(\mathbf{x}) = \mathbf{w}\mathbf{x}$, $\phi'(\mathbf{x}) = \mathbf{w}'\mathbf{x}$

$$\mathbf{G}_{ij}^{sim} = \frac{\exp F(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{j=1}^{\mathrm{N}} \exp F(\mathbf{x}_i, \mathbf{x}_j)}$$

Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Building Graphs – Spatiotemporal graph

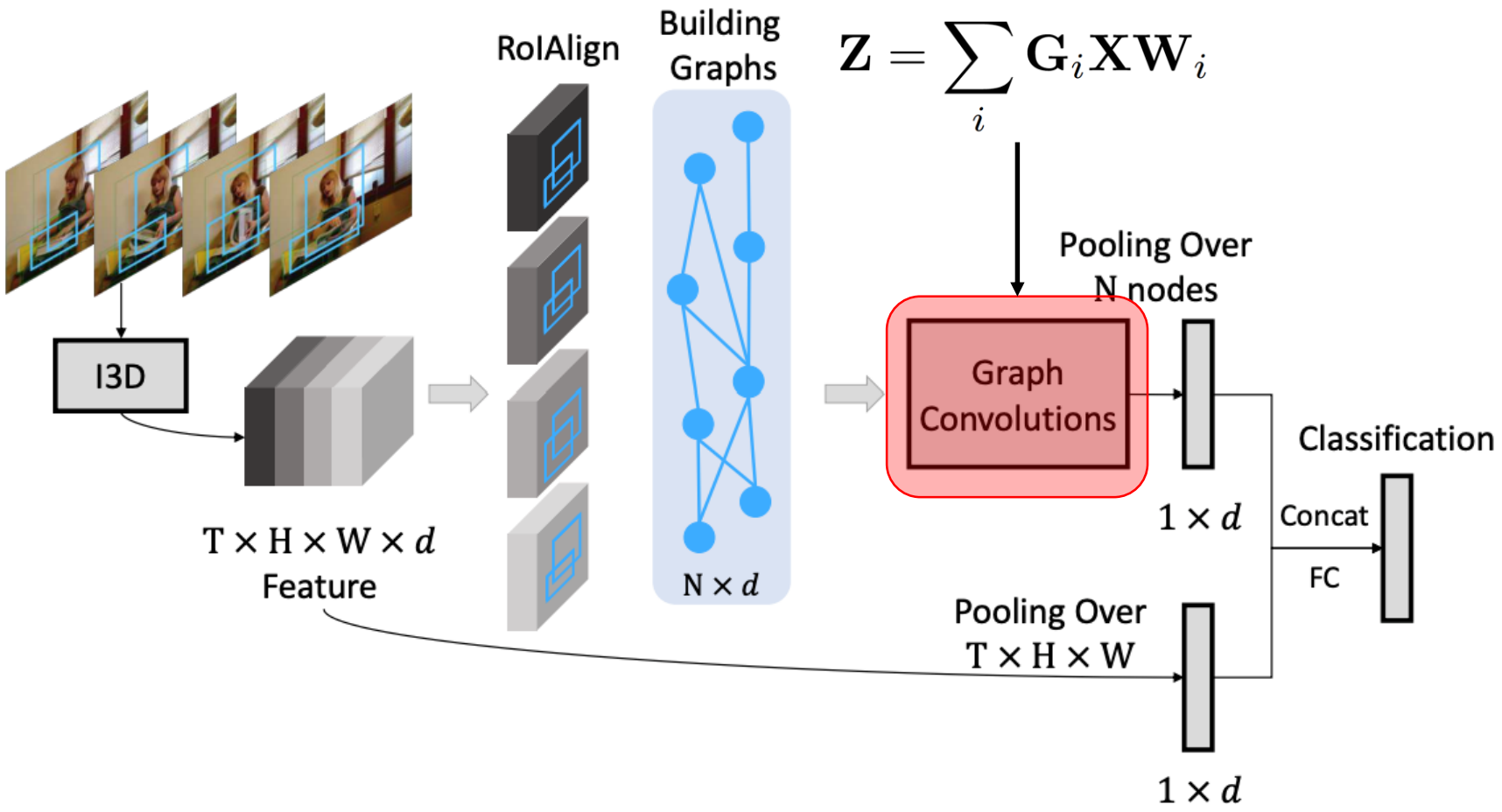
- Use spatio-temporal relation between region proposals.
- Two directed graph (forward graph, backward graph).
- If a proposal at t has an overlap with a proposal at $t+1$, connect them (vice versa in backward graph).



$$\mathbf{G}_{ij}^{front} = \frac{\sigma_{ij}}{\sum_{j=1}^{\mathrm{N}} \sigma_{ij}},$$

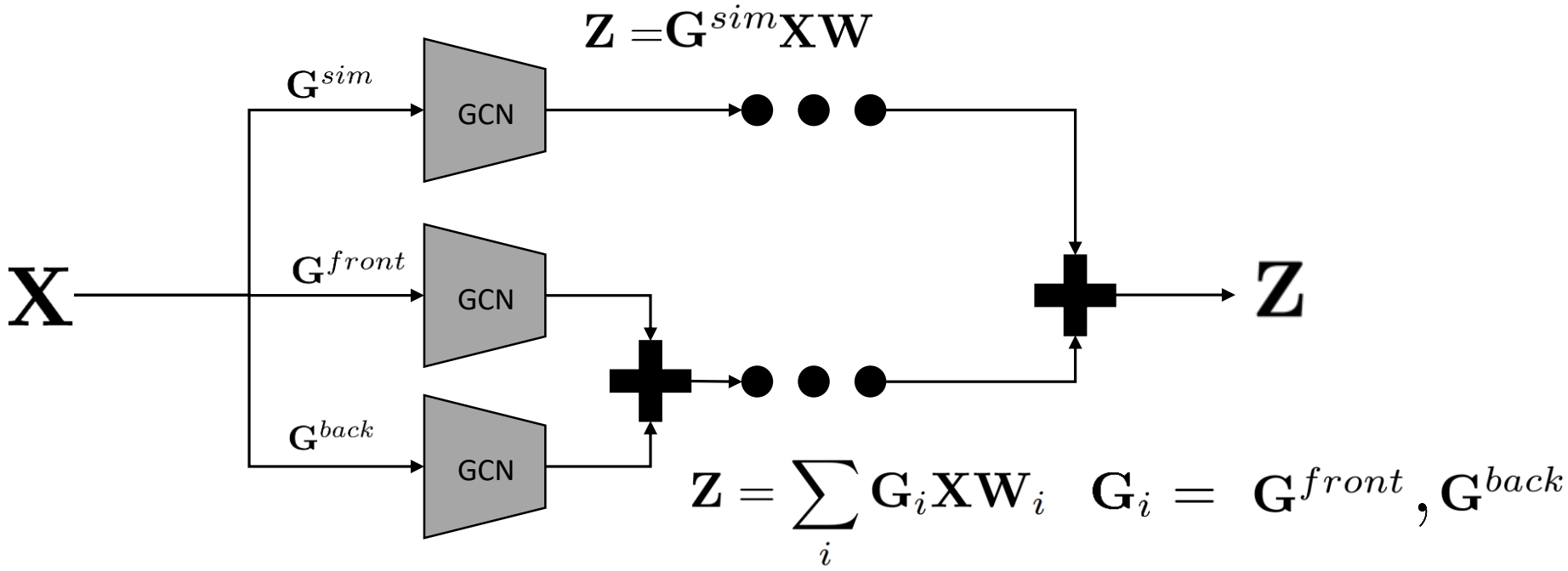
σ_{ij} is an IoU of $\mathbf{X}_i$ and $\mathbf{X}_j$

Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.



Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Graph Convolutions



- Features from Similarity graph and from spatiotemporal graphs are added in the last layer.
  - They are independent until the last layer.

Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Experiments

- ResNet-50 I3D backbone pretrained on Kinetics.
- RPN with ResNet-50-FPN backbone pre-trained on MSCOCO.
- Sample clips (32 frames, 6fps).
- Dataset
  - Charades: 157 classes, multi-action, 8k train, 1.8k validation.
  - Something-Something: 174 classes, single-action 86k train, 12k validation, 11k test.

Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Experiments

model	backbone	modality	mAP
2-Stream [93]	VGG16	RGB + flow	18.6
2-Stream +LSTM [93]	VGG16	RGB + flow	17.8
Asyn-TF [93]	VGG16	RGB + flow	22.4
MultiScale TRN [36]	Inception	RGB	25.2
I3D [8]	Inception	RGB	32.9
I3D [58]	ResNet-101	RGB	35.5
NL I3D [58]	ResNet-101	RGB	37.5
NL I3D + GCN	ResNet-50	RGB	37.5
I3D + GCN	ResNet-101	RGB	39.1
NL I3D + GCN	ResNet-101	RGB	**39.7**

Charades

model	backbone	*val* top-1	*val* top-5	*test* top-1
C3D [21]	C3D[7]	-	-	27.2
MultiScale TRN [36]	Inception	34.4	63.2	33.6
I3D	ResNet-50	41.6	72.2	-
I3D + GCN	ResNet-50	43.3	75.1	-
NL I3D	ResNet-50	44.4	76.0	-
NL I3D + GCN	ResNet-50	46.1	76.8	45.0

Something-Something

Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Ablation studies

model, R50, I3D	mAP
baseline	31.8
Proposal+AvgPool	32.1
Spatial-Temporal GCN	34.2
Similarity GCN	35.0
Joint GCN	**36.2**

Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Conclusion

- Video as Space-Time Region Graph
  - shows Spatio-Temporal Information from RoIs is helpful for action recognition.
  - presents a novel graph representation with variant relationships between objects in a long range video.
  - first uses a Graph Convolutional Network for reasoning with multiple relation edges on video action recognition.

Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Code

- Since there is no public code, I reproduce the implementation.
- https://github.com/jd730/STRG

Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Experiments

- The architecture of ResNet-50 I3D(left) is quite different from the original ResNet-50 I3D (right)

layer		output size
$conv_1$	5×7×7, 64, stride 1, 2, 2	32×112×112
$pool_1$	1×3×3 max, stride 1, 2, 2	32×56×56
res_2	$\begin{bmatrix} 3\times1\times1, 64 \\ 1\times3\times3, 64 \\ 1\times1\times1, 256 \end{bmatrix} \times3$	32×56×56
$pool_2$	3×1×1 max, stride 2, 1, 1	16×56×56
res_3	$\begin{bmatrix} 3\times1\times1, 128 \\ 1\times3\times3, 128 \\ 1\times1\times1, 512 \end{bmatrix} \times4$	16×28×28
res_4	$\begin{bmatrix} 3\times1\times1, 256 \\ 1\times3\times3, 256 \\ 1\times1\times1, 1024 \end{bmatrix} \times6$	16×14×14
res_5	$\begin{bmatrix} 3\times1\times1, 512 \\ 1\times3\times3, 512 \\ 1\times1\times1, 2048 \end{bmatrix} \times3$	16×14×14
global average pool, fc		1×1×1

Table 1. Our baseline ResNet-50 I3D model. We use T×H×W to represent the dimensions of filter kernels and 3D output feature maps. For filter kernels, we also have number of channels following T×H×W. The input is in 32×224×224 dimensions and the residual blocks are shown in brackets.

Model	Block	conv1	conv2_x		conv3_x		conv4_x		conv5_x		
			F	*N*	*F*	*N*	*F*	*N*	*F*	*N*	
ResNet-{18, 34}	Basic	conv, 7×7×7, 64, temporal stride 1, spatial stride 2	64	{2, 3}	128	{2, 4}	256	{2, 6}	512	{2, 3}	global average pool, *C*-d fully-connected, softmax
ResNet-{50, 101, 152, 200}	Bottleneck		64	3	128	{4, 4, 8, 24}	256	{6, 23, 36, 36}	512	3	
Pre-act ResNet-200	Pre-act		64	3	128	24	256	36	512	3	
WRN-50	Bottleneck		128	3	256	4	512	6	1024	3	
ResNeXt-101	ResNeXt		128	3	256	24	512	36	1024	3	
DenseNet-{121, 201}	DenseNet		64	{6, 6}	128	{12, 12}	256	{24, 48}	{512, 896}	{16, 32}	

Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh, "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?", CVPR, 2018.

Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Change parameters

- We use ImageNet pre-trained ResNet-50-I3D.
  - Same architecture of the paper.
- We change several hyperparameters to shorten the training time.
  - batch size: 8 -> 32
  - RPN sampling interval: 2 -> 16 (due to the architecture difference)

Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Training Code (including RPN sampling)

```
    for i, (inputs, targets) in enumerate(data_loader):
        data_time.update(time.time() - end_time)
        targets = targets.to(device, non_blocking=True)
        if rpn is not None:
            '''
                There was an unexpected CUDNN_ERROR when len(rpn_inputs) is
                decrased.
            '''
            N, C, T, H, W = inputs.size()
            if i == 0:
                max_N = N
            # sample frames for RPN
            sample = torch.arange(0,T,det_interval)
            rpn_inputs = inputs[:,:,sample].transpose(1,2).contiguous()
            rpn_inputs = rpn_inputs.view(-1,C,H,W)
            if len(inputs) < max_N:
                print("Modified from {} to {}".format(len(inputs), max_N))
                while len(rpn_inputs) < max_N * (T // det_interval):
                    rpn_inputs = torch.cat((rpn_inputs, rpn_inputs[:(max_N-len(inputs))*(T//det_interval)]))
            with torch.no_grad():
                proposals = rpn(rpn_inputs)
            proposals = proposals.view(-1,T//det_interval,nrois,4)
            if len(inputs) < max_N:
                proposals = proposals[:len(inputs)]
            outputs = model(inputs, proposals.detach())
            # update to the largest batch_size
            max_N = max(N, max_N)
        else:
            outputs = model(inputs)
        loss = criterion(outputs, targets)
        acc = calculate_accuracy(outputs, targets)

        losses.update(loss.item(), inputs.size(0))
        accuracies.update(acc, inputs.size(0))

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```


Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Model (strg.py)

```
70      def forward(self, inputs, rois=None):
71          features = self.extract_feature(inputs)
72          features = self.reducer(features) # N C T H W
73          pooled_features = self.avg_pool(features).squeeze(-1).squeeze(-1).squeeze(-1)
74          N, C, T, H, W = features.shape
75
76          rois_list = rois.view(-1, self.nrois, 4)
77          rois_list = [r for r in rois_list]
78
79          features = features.transpose(1,2).contiguous().view(N*T,C,H,W)
80          rois_features = self.roi_align(features, rois_list)
81          rois_features = self.max_pool(rois_features)
82          rois_features = rois_features.view(N,T,self.nrois,C)
83          gcn_features = self.strg_gcn(rois_features, rois)
84
85          features = torch.cat((pooled_features, gcn_features), dim=-1)
86          outputs = self.classifier(features)
87
88          return outputs
```


Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Graph Generation

- module/roi_graph.py (right)
- rgcn_models.py (left)

```python
def sim_graph(self, features):
    sim1 = self.sim_embed1(features)
    sim2 = self.sim_embed2(features)
    sim_features = torch.matmul(sim1, sim2.transpose(1,2)) # d x d mat.
    sim_graph = F.softmax(sim_features, dim=-1)
    return sim_graph
```

```python
def get_st_graph(rois, threshold=0):
    B, T, N, _ = rois.size()

    M = T*N
    front_graph = torch.zeros((B,M,M))

    if M ==0 :
        return front_graph, front_graph.transpoes(1,2)
    areas = (rois[:,:,:,3] - rois[:,:,:,1] + 1) * (rois[:,:,:,2] - rois[:,:,:,0] + 1)

    for t in range(T-1):
        for i in range(N):
            ious = get_iou(rois[:,t,i], rois[:,t+1], areas[:,t,i:i+1], areas[:,t+1])
            ious[ious < threshold] = 0
            front_graph[:, t*N+i, (t+1)*N:(t+2)*N] = ious

    back_graph = front_graph.transpose(1,2)

    # Normalize
    front_graph = front_graph / front_graph.sum(dim=-1, keepdim=True)
    back_graph = back_graph / back_graph.sum(dim=-1, keepdim=True)
    # NaN to zero
    front_graph[front_graph != front_graph] = 0
    back_graph[back_graph != back_graph] = 0

    return front_graph, back_graph
```


Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Graph Convolution (rgcn_models.py)

```python
    def forward(self, rois_features, rois):
        front_graph, back_graph = get_st_graph(rois)

        front_graph = front_graph.to(rois.device).detach()
        back_graph = back_graph.to(rois.device).detach()

        B, T, N, C = rois_features.size()
        N_rois = T*N
        rois_features = rois_features.view(B, N_rois, -1)
        sim_graph = self.sim_graph(rois_features).detach()
        sim_gcn = self.sim_GCN(rois_features, sim_graph)
        st_gcn = self.st_GCN(rois_features, front_graph, back_graph)
        gcn_out = sim_gcn + st_gcn
        gcn_out = gcn_out.mean(1)
        gcn_out = self.dropout(gcn_out)
        return gcn_out
```

```python
    def st_GCN(self, input, front_graph, back_graph=None):
        input = input.squeeze(2)
        out = F.relu(self.st_gc1(input,front_graph))
        if self.separate_fb:
            out += F.relu(self.st_gc1_back(input,back_graph))
#        out = self.dropout(out)

        out2 = F.relu(self.st_gc2(out,front_graph))
        if self.separate_fb:
            out2 += F.relu(self.st_gc2_back(out,back_graph))
        out = out2
#        out = self.dropout(out2)

        out2 = F.relu(self.st_gc3(out,front_graph))
        if self.separate_fb:
            out2 += F.relu(self.st_gc3_back(out,back_graph))
        return out2


    def sim_GCN(self, input, adj):
        out = F.relu(self.sim_gc1(input,adj))
#        out = self.dropout(out)
        out = F.relu(self.sim_gc2(out,adj))
#        out = self.dropout(out)
        out = F.relu(self.sim_gc3(out,adj))
        return out
```


Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Running

Something-Something-v1

First of all, we need to train backbone network (ResNet-50-I3D) for 100 epochs with learning rate as 0.00125 (decayed at 90 epoch to 0.000125) The original batchsize is 8 but in this implementation, we use 32 to reduce the training time.

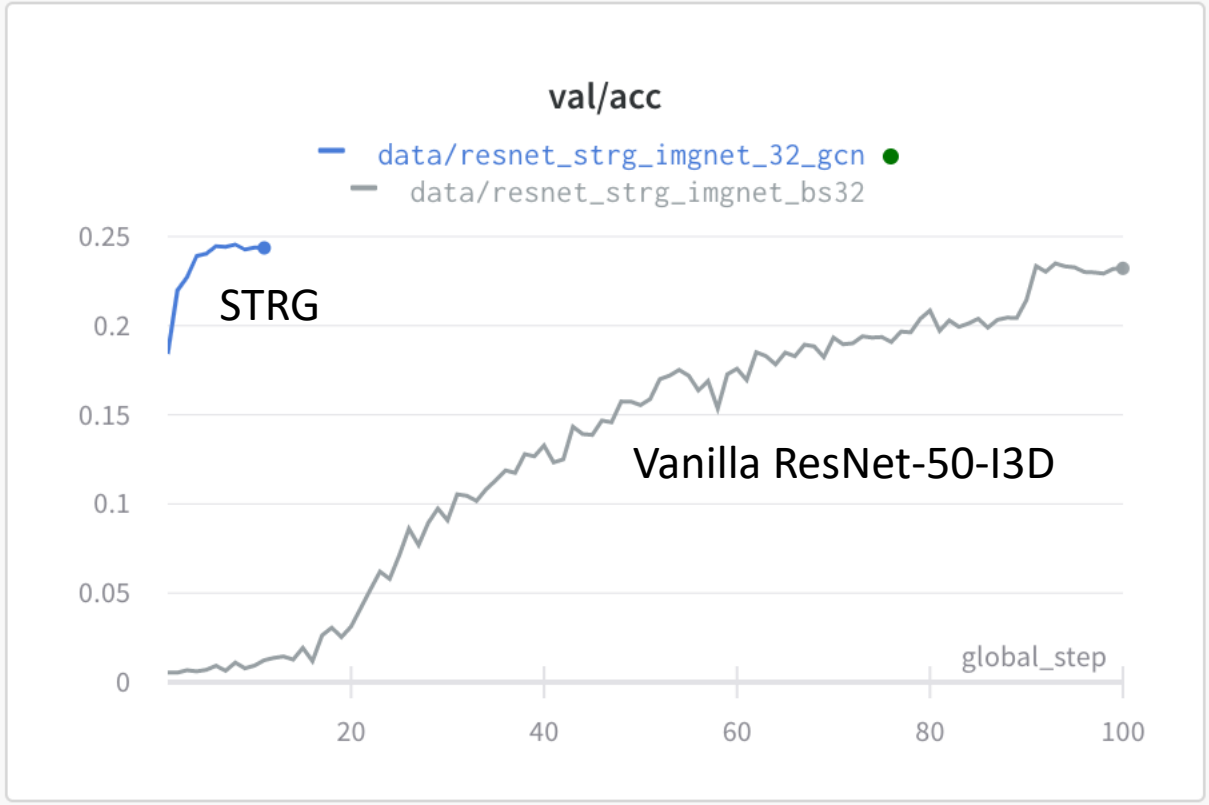
```
python main.py --root_path data --video_path data/something/v1/img --annotation_path sthv1.json \
--result_path resnet_strg_imgnet_bs32 --dataset somethingv1 --n_classes 174 --n_pretrain_classes 700 \
--ft_begin_module fc --tensorboard --wandb --conv1_t_size 5 --learning_rate 0.00125 --sample_duration 32 \
--n_epochs 100 --multistep_milestones 90 --model resnet_strg --model_depth 50 --batch_size 32 \
--n_threads 8 --checkpoint 1
```

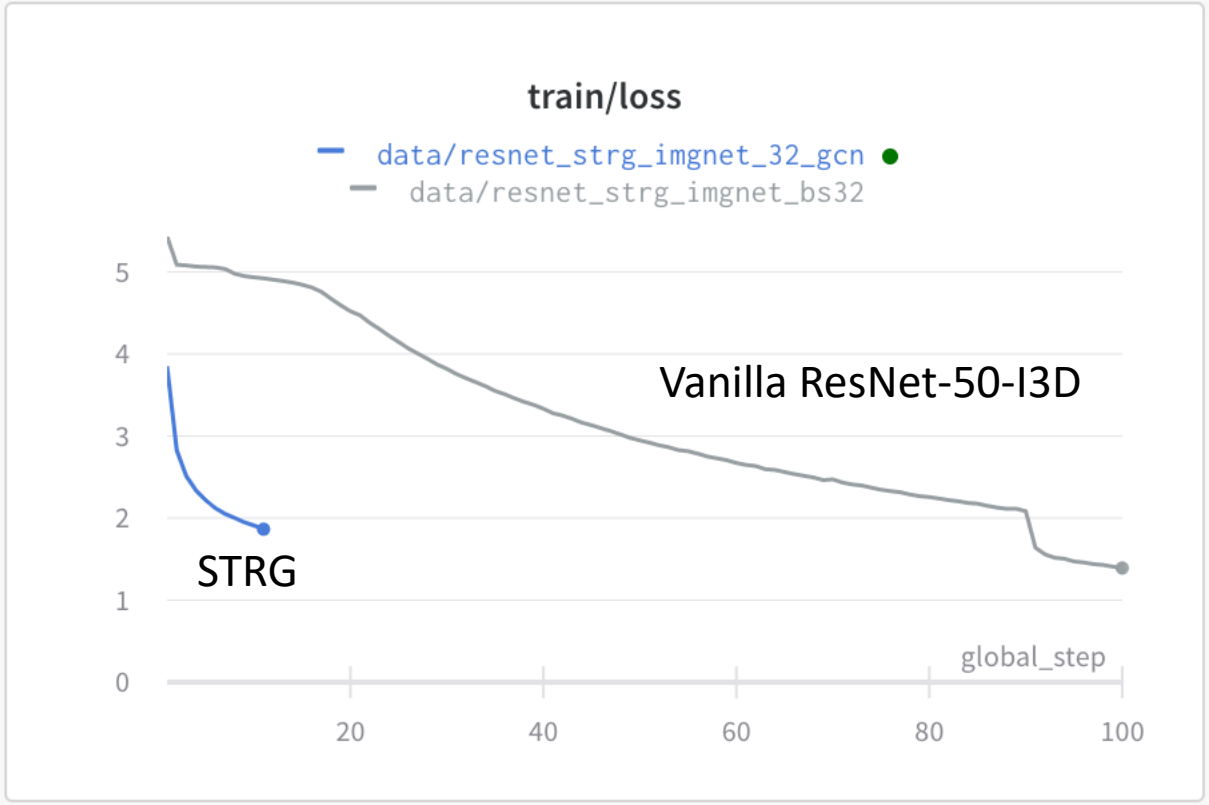
Then, we need to train with GCN module until 30 epochs with learning rate as 0.000125.

```
python main.py --root_path data --video_path data/something/v1/img --annotation_path sthv1.json \
--result_path resnet_strg_imgnet_32_gcn --dataset somethingv1 --n_classes 174 --n_pretrain_classes 174 \
--ft_begin_module fc --tensorboard --wandb --conv1_t_size 5  --learning_rate 0.000125 \
--sample_duration 32 --n_epochs 30 --model resnet_strg --model_depth 50 --batch_size 32 \
--nrois 10 --det_interval 2 --strg \
--n_threads 8 --checkpoint 1 --pretrain_path resnet_strg_imgnet_bs32/save_100.pth
```


Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Results



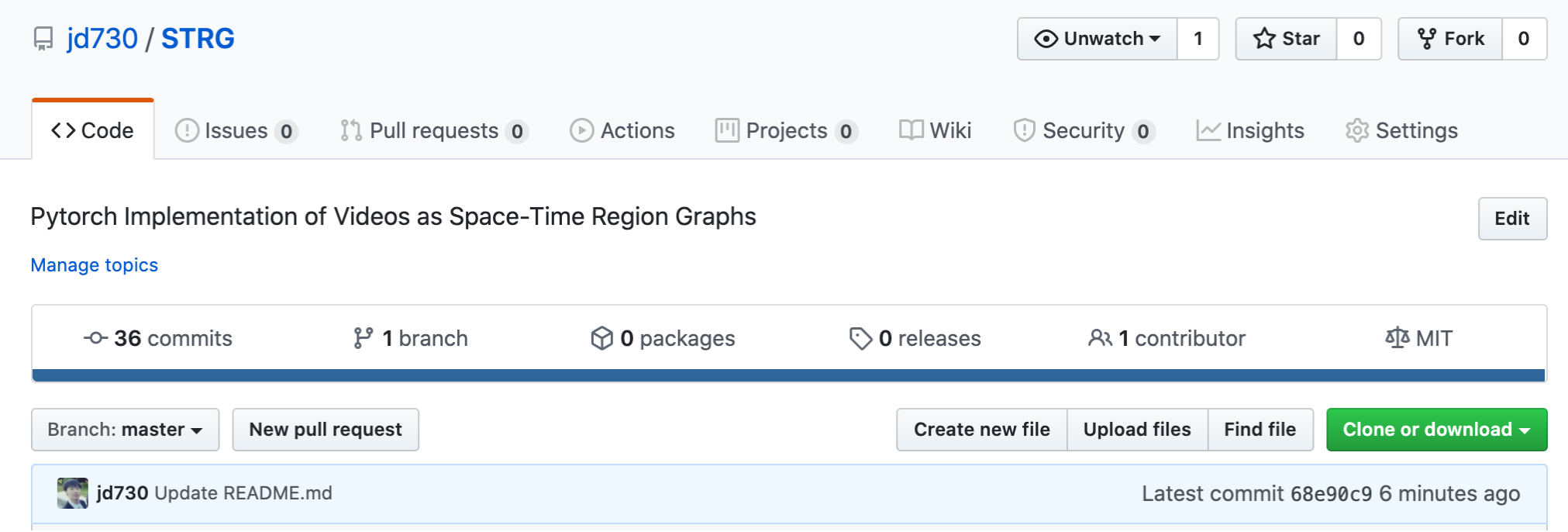


Model name	ResNet-50-I3D	ResNet-50-I3D + STRG
Top-1 Accuracy	23.2%	24.5%

Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

For more details,

- Please refer to as the README.md in https://github.com/jd730/STRG



Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.

Commit Log

Commits on Jun 19, 2020

Update README.md jd730 committed 7 minutes ago	Verified	68e90c9
[change] default value of interval jd730 committed 8 minutes ago		50dcfa5
Update README.md jd730 committed 12 minutes ago	Verified	2717315
Update README.md jd730 committed 12 minutes ago	Verified	00db0d0
Update README.md jd730 committed 14 minutes ago	Verified	621699a
[Add] flags and refactors jd730 committed 41 minutes ago		d29da77

Commits on Jun 13, 2020

[fix] bug when N < max_N * 0.5 jd730 committed 6 days ago	47b418c
[fix] training jd730 committed 6 days ago	439d8eb
[fix] validation error while using RPN jd730 committed 6 days ago	b98d2a6
[change] resume position jd730 committed 6 days ago	b1d7e08
[change] RPN return type jd730 committed 6 days ago	2e8e667

Commits on Jun 12, 2020

[change] load_imagenet_pretrained to generalize jd730 committed 7 days ago	cde0777
[add] imagenet pre-trained weight on resnet_strg jd730 committed 7 days ago	c23bf80

Commits on Jun 11, 2020

[add] assert resnet_strg jd730 committed 8 days ago		2c46f62
[add] resnet_strg jd730 committed 8 days ago		a54f362
[change] train jd730 committed 8 days ago		f6c923c
[add] freeze bn jd730 committed 8 days ago		08cfdb8
[change] interval and architecture jd730 committed 8 days ago		cc60e54
[add] tensorboardX jd730 committed 8 days ago		1bb8445
[add] h5py jd730 committed 8 days ago		b7b8f08
--no-edit jd730 committed 8 days ago		c09ac65
[add] freeze BN jd730 committed 8 days ago		8da4396
[fix] validation jd730 committed 8 days ago		0c69356
Update README.md jd730 committed 8 days ago	Verified	f912e63
[add] vid2img for sthv2 jd730 committed 8 days ago		8f5605a
[add] rpn to val_epoch jd730 committed 8 days ago		4906be5
[add] STRG on validation jd730 committed 8 days ago		1da4ecb
[add] wandb jd730 committed 8 days ago		2e8df06
[add] STHv1 jd730 committed 8 days ago		f856f29
[add] STRG flag jd730 committed 8 days ago		a7785d6
[add] STRG that can run jd730 committed 8 days ago		47f21e4

Commits on Jun 10, 2020

[add] roi_graph jd730 committed 8 days ago	2966200
[add] STRG base code jd730 committed 9 days ago	1178e06
[add] RPN jd730 committed 9 days ago	d10d7f3
[add] STHv2 jd730 committed 9 days ago	0a020e8

Commits on Jun 10, 2020

[init] from https://github.com/kenshohara/3D-ResNets-PyTorch jd730 committed 9 days ago	30e8f99

Wang, Xiaolong, and Abhinav Gupta. "Videos as space-time region graphs." *ECCV*. 2018.