**Paper Analysis:** Spatial-Temporal Graph Convolutional Networks for Sign Language Recognition-

Cleison Correia de Amorim, David Macˆedo, Cleber Zanchettin

**Professor:** Ashutosh Modi

**Mentor:** Abhinav Joshi

**Presenter:** Shashwat Gupta (200923)

**What?**

1. STGNNs to capture dynamics of the signs in two dimensions - spatial and temporal

2. A New dataset of human skeletons for sign language based on ASLLVD to contribute to future related studies.

**Why?**

[0],[6]

Salient Benefits of ST-GCN:

* It is more natural (realistic / real-worldly) to represent the skeletons as graphs rather than as 2D/3D images
* Automatic feature extraction as opposed to hand-crafted parts or rules (in deep learning methods like – temporal CNNs, RNNs) to analyse spatial patterns

Existing approaches for skeleton based action recognition have verified the effectiveness of introducing body parts in the modeling. We argue that the improvement is largely due to that parts restrict the modeling of joints trajectories within “local regions” compared with the whole skeleton, thus forming a hierarchical representation of the skeleton sequences. In tasks such as image object recognition, the hierarchical representation and locality are usually achieved by the intrinsic properties of convolutional neural networks

* Earlier methods using joined coordinates for indivisual time steps to form feature vectors (analogue of downsampling in UNets) missed spatial relationships among joints (which play important role in actions e.g. similar motions of hands in different relative locations)
* The ST-GCN model is quite flexible.

Related Work: (Need for Temporal Analysis)

A lot of work done in STL focusses on static aspects. To Analyse the dynamic aspects, we need temporal analysis.

([4], [5]: <https://sci-hub.hkvisa.net/10.1109/dicta.2017.8227483>, [7])

* Sensors and more powerful hardware (intrusive)

1. Touch based – inhibit naturalness, extensive calibration
2. Vision/Untouched (common – using depth maps)

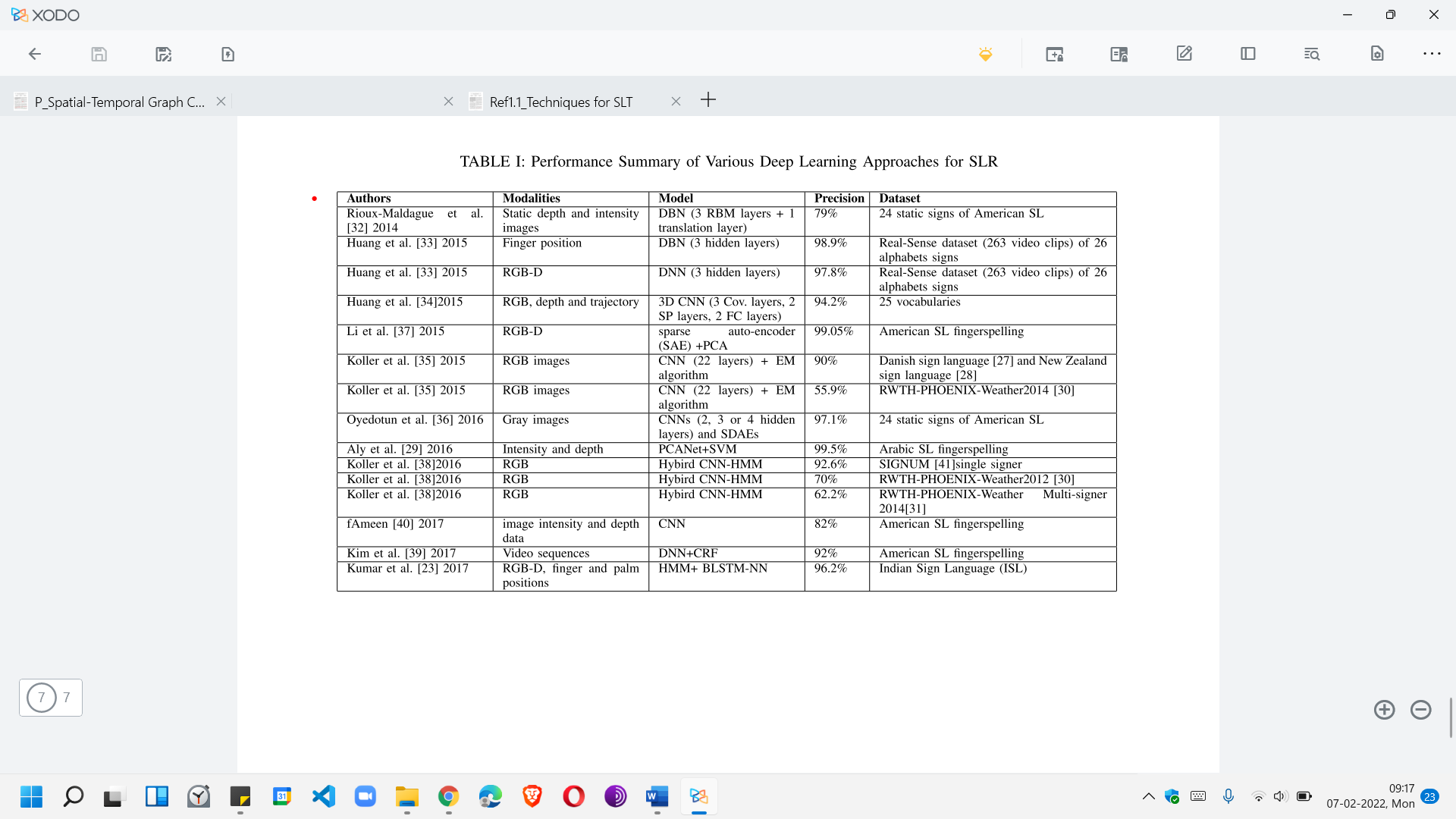
* (ML – CV):

Framework: Pre-processing (removing noise and hand localisation) -> sign gesture representation -> feature extraction -> Classification -> Output

Conventional Methods:

* 2D features include – SIFT (Scale Invariant Feature Transform), HOG (Histogram of Gradients), HOF (Histogram of Optical Flow), STIP (Spatial Temporal Interest Points), and kernel descriptors to pre-process images for feature extraction. (good performance only for the case of single and clear object recognition)
* 3D/4D space of time (HON4D – Histogram of Oriented 4 D Normals) and Random Occupancy Pattern (ROP) – occlusions
* Moreover, to address the issue of noise and occlusions in the depth maps, Space-Time Occupancy Patterns (STOP) was proposed to characterize the 4D space-time patterns of human gestures to leverage the spatial and temporal contextual information while allowing for intra-class variations.
* motion history templates based on multiscale, pyramid motion history templates while considering multiple temporal scales and multiple levels of spatial grids. Commonly used classifiers have been seen as template matching, dictionary learning, bag of visual words [9, 10], Support Vector Machines (SVMs) [11, 12], and Hidden Markov Models (HMMs)[13]
* [8], [9] Convolutional Neural Networks (CNN), Recurrent Neural Networks and Temporal Residual Networks - interesting results (dep on dataset and other combined models) (90% accuracy depending on dataset) There are still some variations as 3D CNNs, the combination with other models such as Inception or the Regions of Interest applications [10], [11], [12].
* Recurrent Neural Networks and Temporal Residual Networks also obtained interesting results in the same purpose [13], [14].

Recent Advancements in Deep Learning for Skeletal Recognition:

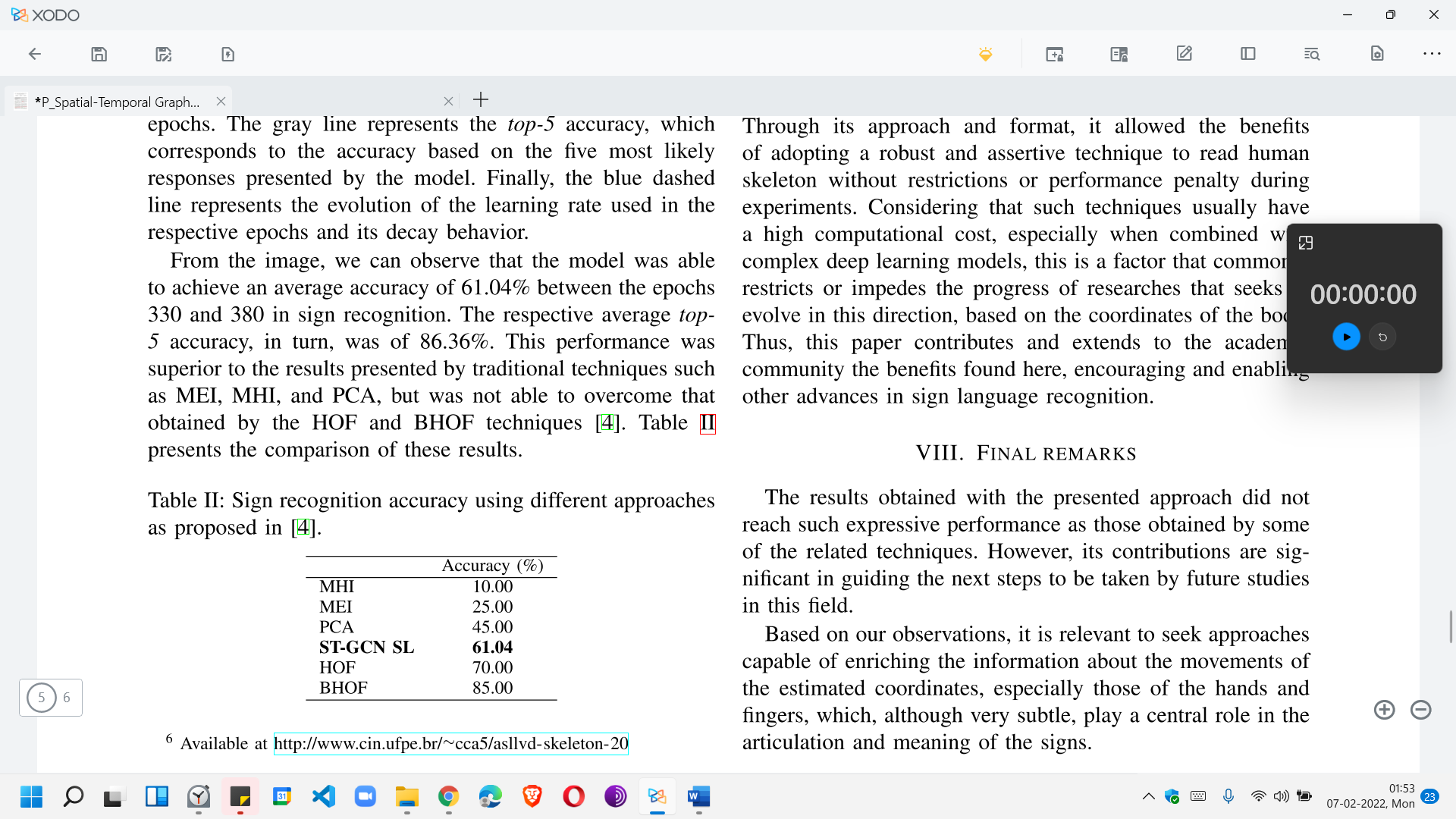
****

**Challenge** = Not much work on dynamic aspects (movements, articulation between different body parts and non-manual expressions)

**STGNN** = (body parts and disregards environments) + (spatial + temporal dimensions)

Experimental Results for ST-GCN:

[0]

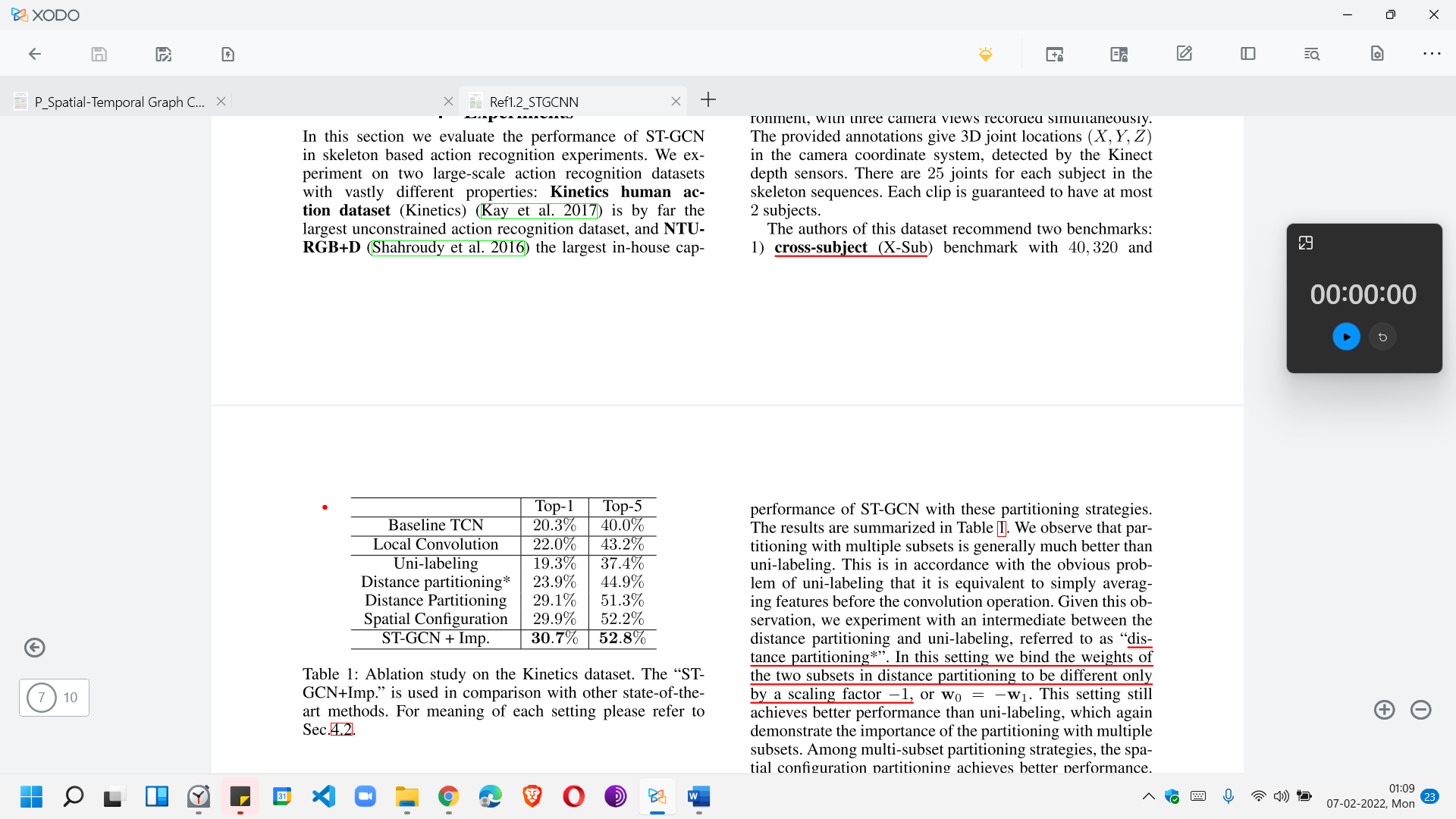


Indeed, the application of consecutive steps in BHOF for optical flow extraction, color map creation, block segmentation and generation of histograms from them were able to ensure that more enhanced features about the hand movements were extracted favoring its sign recognition performance. This technique is derived from HOF and differs only by the approach of focusing on the hands of individuals while calculating the optical flow histogram.

Primitive methods like MEI and MHI detect the movements and their intensity from the difference between the consecutive frames of actions. They are not able to differentiate individuals or to focus on specific parts of their body, causing movements of any nature considered equivalently. The PCA, in turn, adds the ability to reduce the dimensionality of the components based on the identification of those with more significant variance and that, consequently, is more relevant for movement detection in the frames.

[6]

Spatial Information:

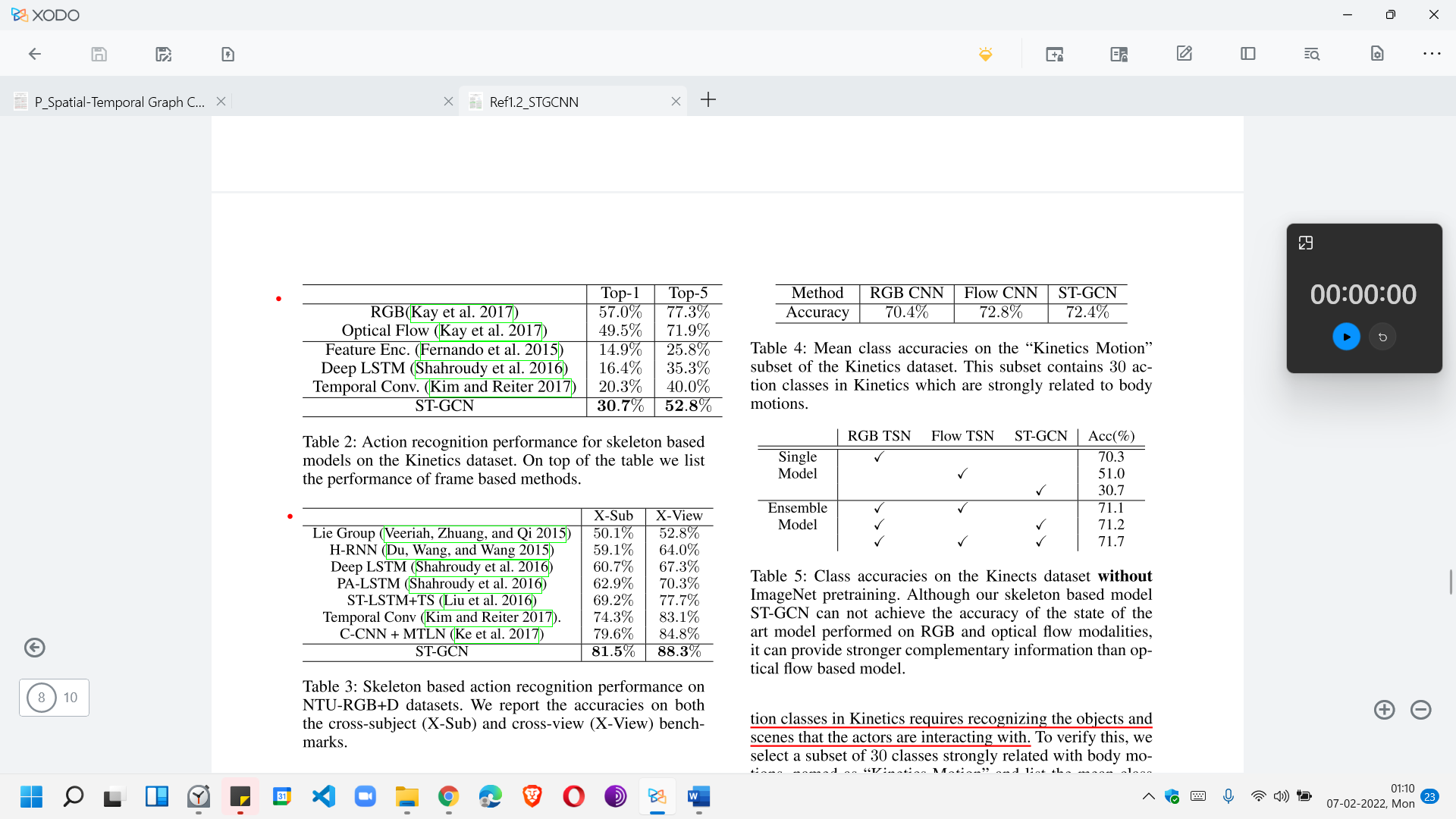


Baseline TCN equivalent to spatial temporal graph convolution with unshared weights on a fully connected joint graph. So the major difference between the baseline model and ST-GCN models are the sparse natural connections (spatial information) and shared weights in convolution operation.

Intermediate model (“local convolution”) = sparse joint graph as ST-GCN, but use convolution filters with unshared weights.

Other SOTA Techiniques:

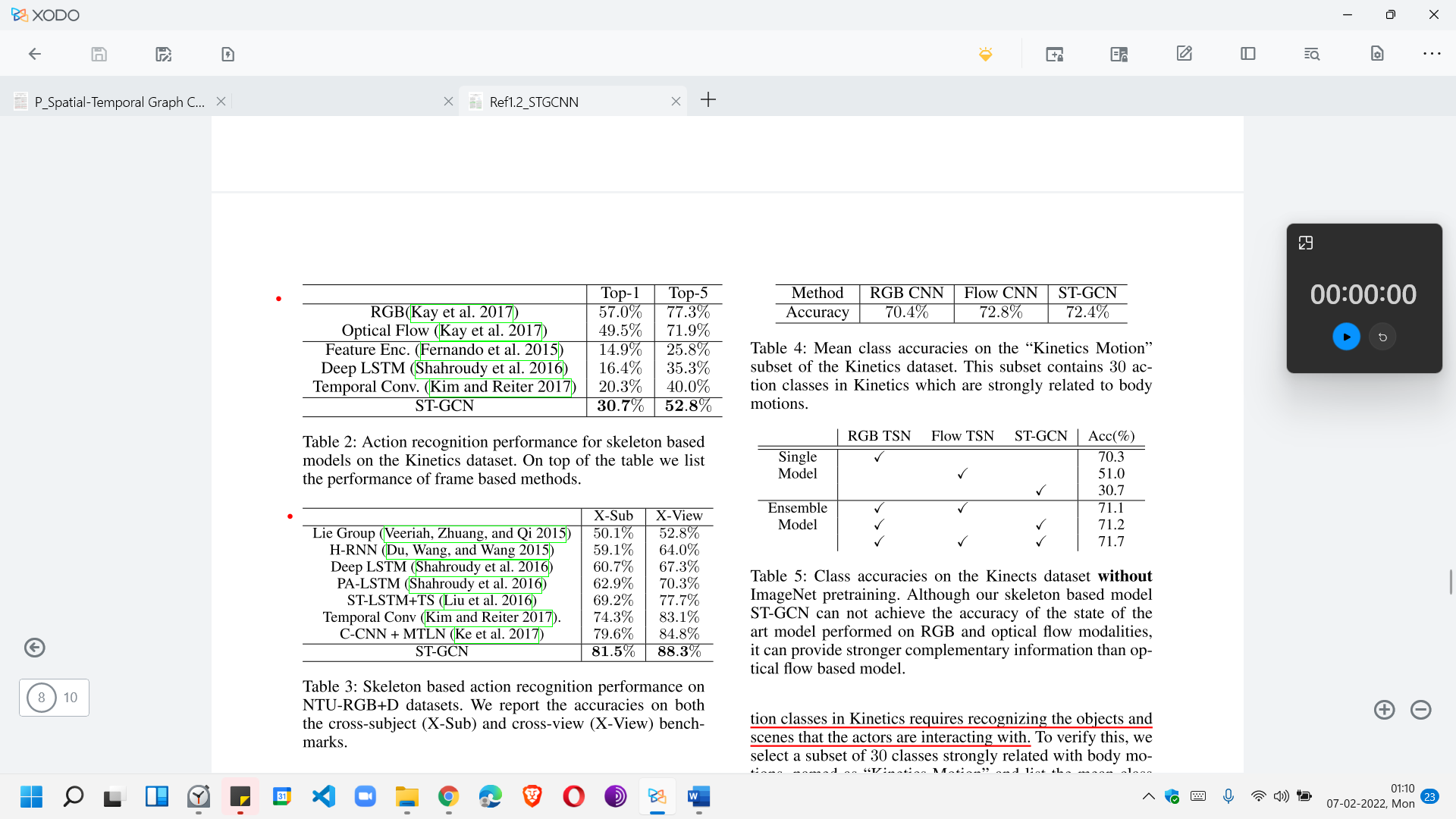
1. Kinetics DataSet: 2D raw clips (hand held camera) detected with DNN



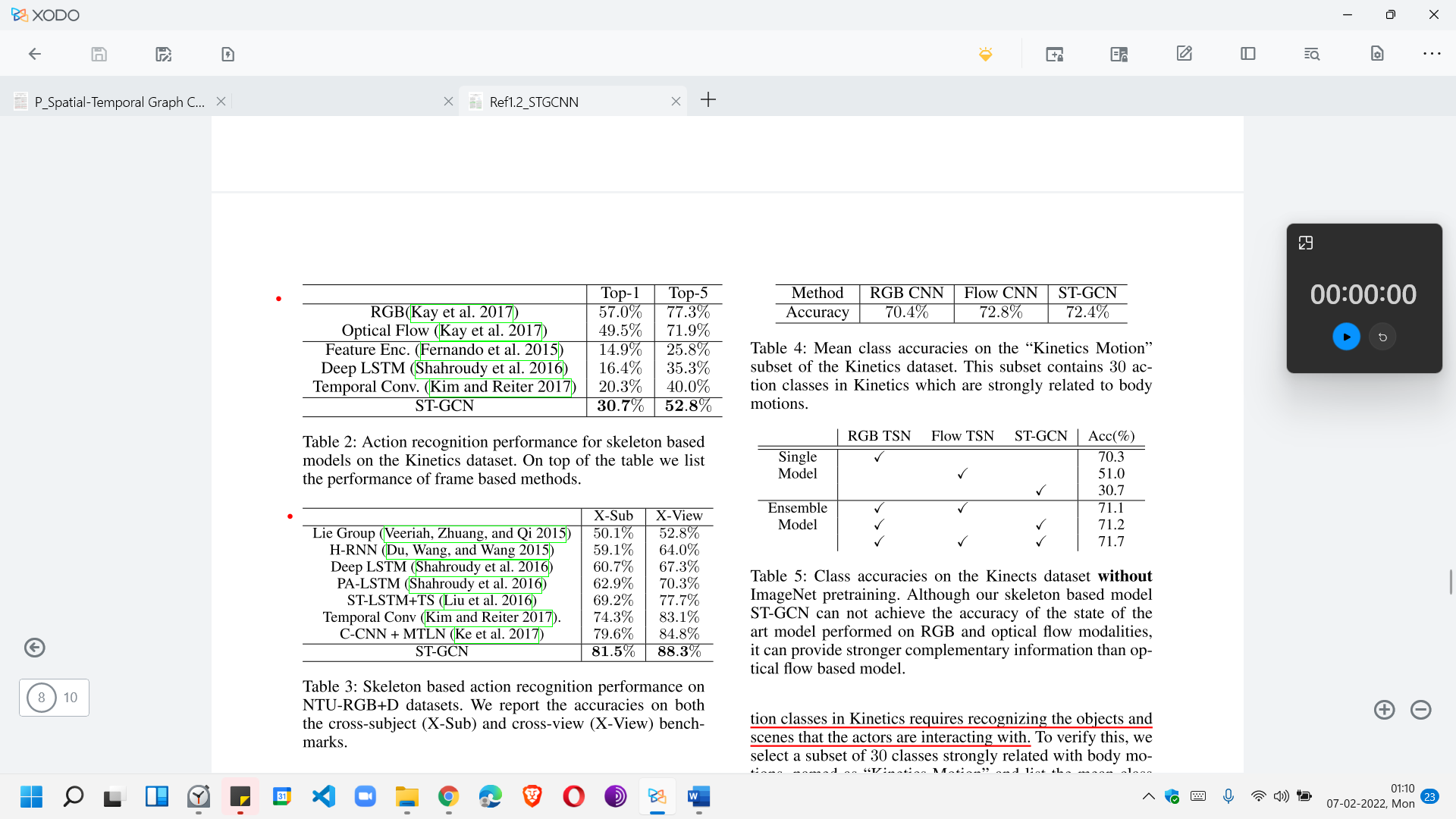
1. NTU-RGB+D DataSet: 3D joint location annotations x,y,z) action clips detected with Kinnecte Depth Sensor (fixed camera)

2 benchmarks:

* Cross -Sub(X-View): training clips come from one subset of actors and the models are evaluated on clips from the remaining actors
* Crossview(X-View): Training clips in this setting come from the camera views 2 and 3, and the evaluation clips are all from the camera view 1



Kinetics the accuracies of skeleton based methods are inferior to video frame based. (Table 1) We argue that this is due to a : lot of action classes in Kinetics requires recognizing the objects and scenes that the actors are interacting with . Picking 30 classes strongly related with body motions, named as “Kinetics-Motion” and list the mean class accuracies of skeleton and frame based models We can see that on this subset the performance gap is much smaller.



**How?**

**1. ST-CGCN**

[6] (ST-GCN Original Paper - S. Yan, Y. Xiong, and D. Lin):

Graph:

Particularly, we construct an undirected spatial temporal graph G = (V, E) on a skeleton sequence with N joints and T frames featuring both intra-body and inter-frame connection.

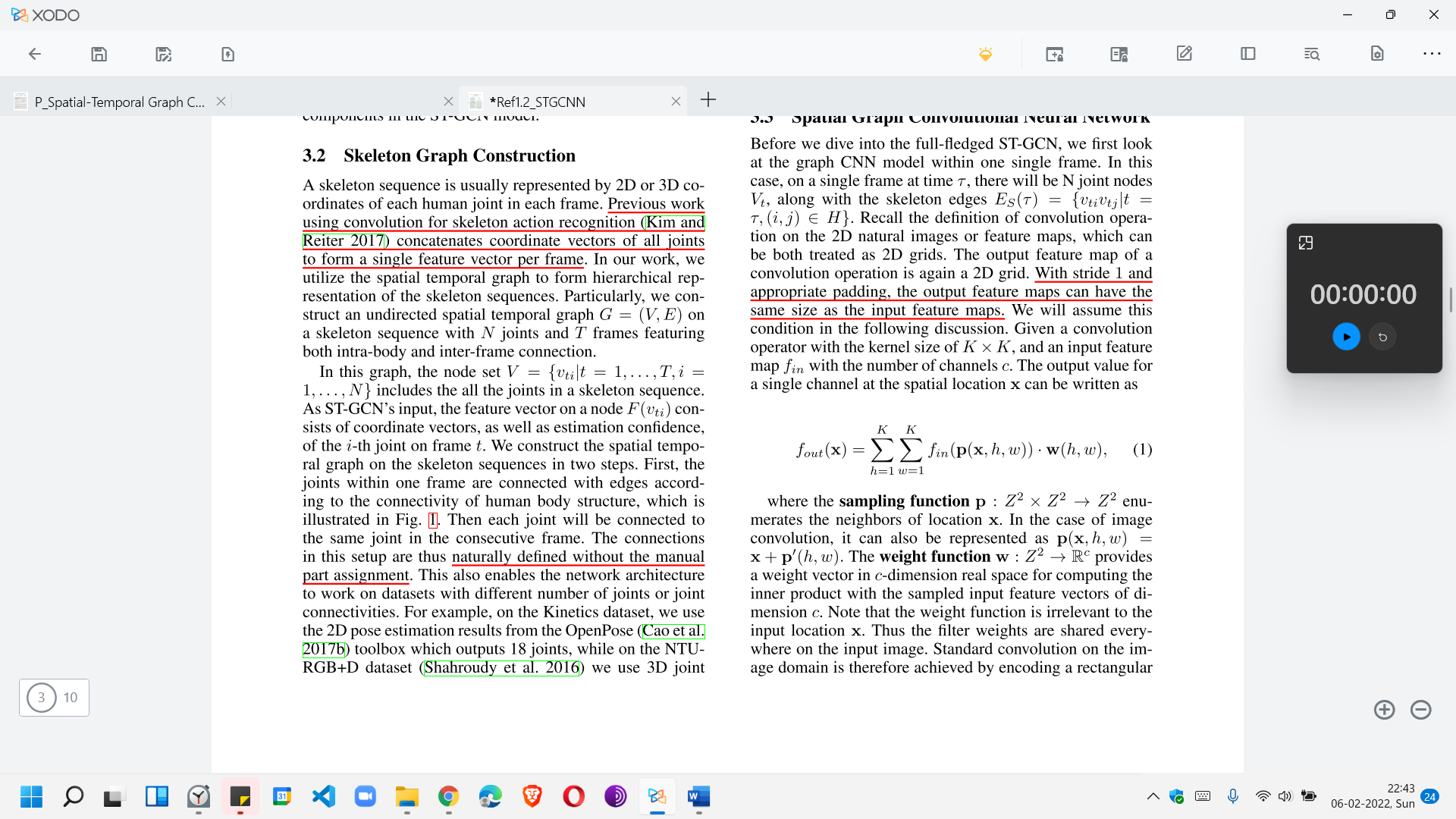
In this graph, the node set V = {vti|t = 1, . . ., T, i =1, . . ., N} includes the all the joints in a skeleton sequence. As ST-GCN’s input, the feature vector on a node F(vti) consists of coordinate vectors, as well as estimation confidence, of the i-th joint on frame t.

E is composed of two subsets:

* intra-skeleton connection at each frame (ES = {vti,vtj|(i, j) ∈ H}, where H is the set of naturally connected human body joints.
* inter-frame edges connecting same joints EF = {vti,v(t+1)i}.

Convolution:

2D Image Convolution:



p=sampling function (neighbours), x in centre, w() is weight in convolutional filter, f is the feature map

Modification for Graph:

The principle of constructing GCNs on graph generally follows two streams:

1) the spectral perspective, where the locality of the graph convolution is considered in the form of spectral analysis

2) the spatial perspective, where the convolution filters are applied directly on the graph nodes and their neighbors

We follow the second stream. We limit the conv-filter by D=1

Spatial:

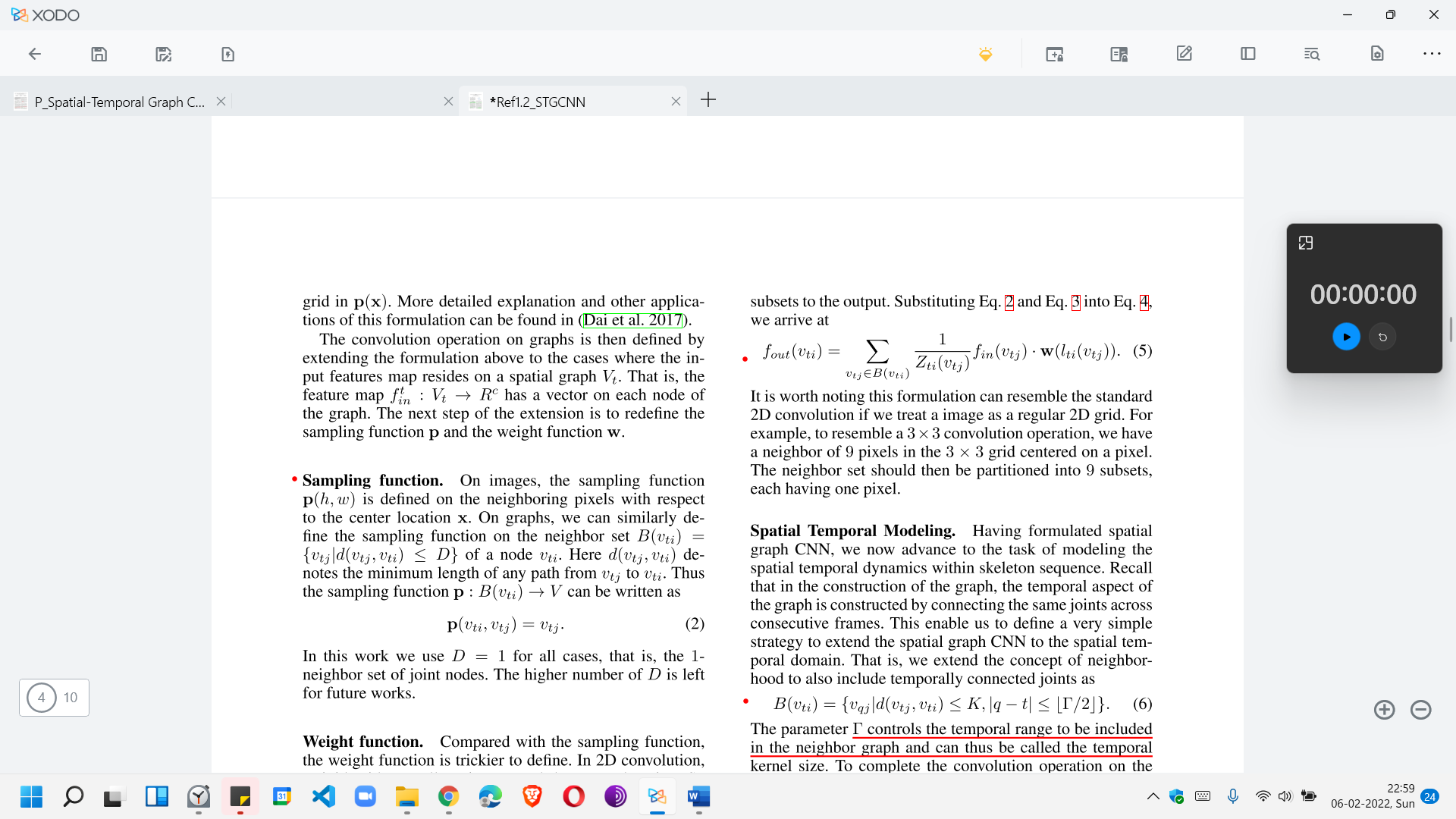
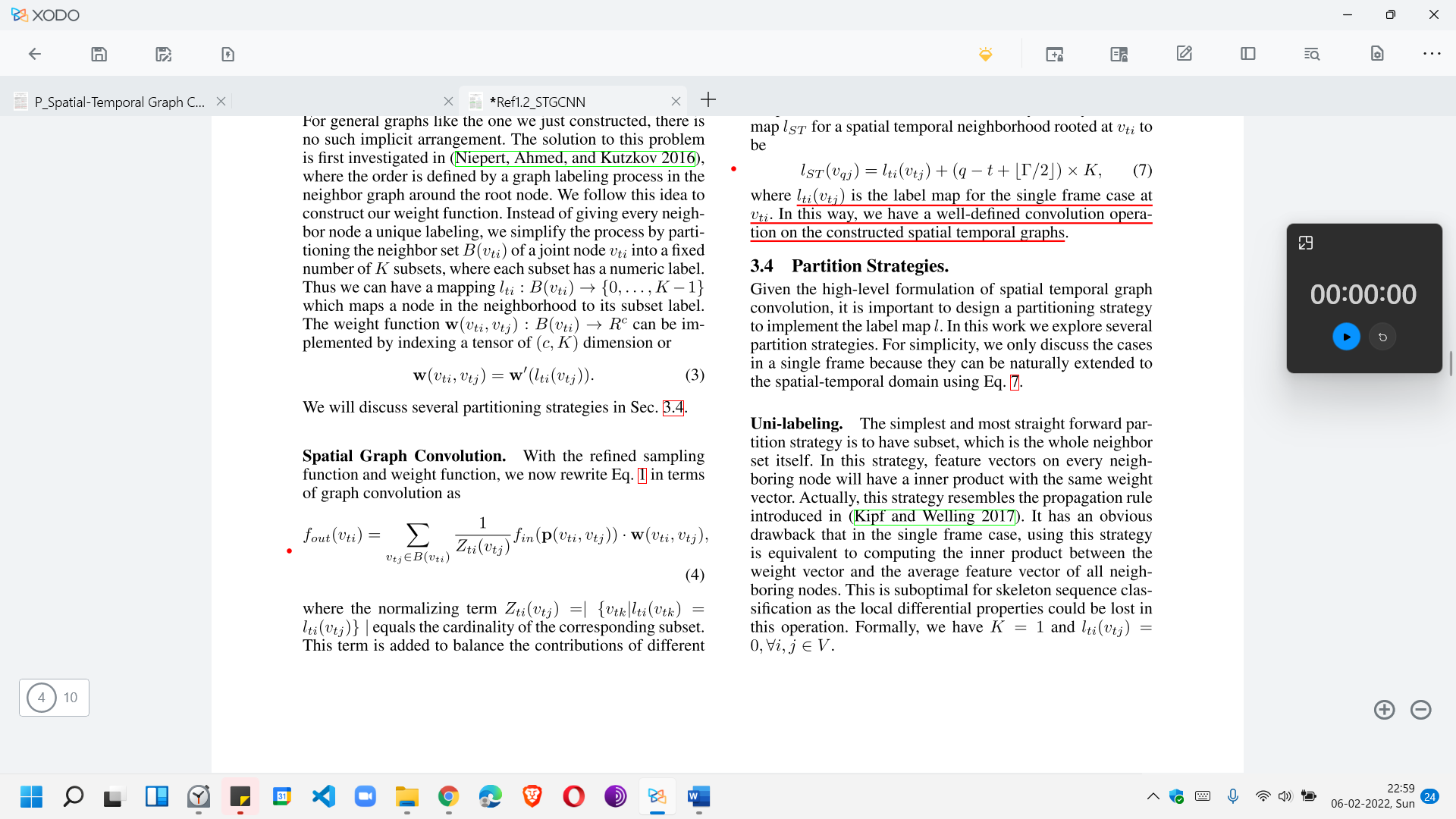
* Feature Map : ftin : Vt → Rc ; f=feature map associated with each vertex of each graph
* Sampling Function : p(vti, vtj) = vtj

(B(vti) = {vtj|d(vtj, vti) ≤ D} of a node vti. Here d(vtj, vti) denotes the minimum length of any path from vtj to vti. Thus the sampling function p : B(vti) → V)

* Weight function : w(vti, vtj) = w(lti(vtj))

(Instead of giving every neighbor node a unique labeling, we simplify the process by **partitioning** the neighbor set B(vti) of a joint node vti into a fixed number of K subsets, where each subset has a numeric label. Thus we can have a label-mapping lti : B(vti) → {0, . . . ,K−1} which maps a node in the neighborhood to its subset label.)

So the convolution is



where Zti(vtj) =| {vtk|lti(vtk) = lti(vtj)} | equals the cardinality of the corresponding subset

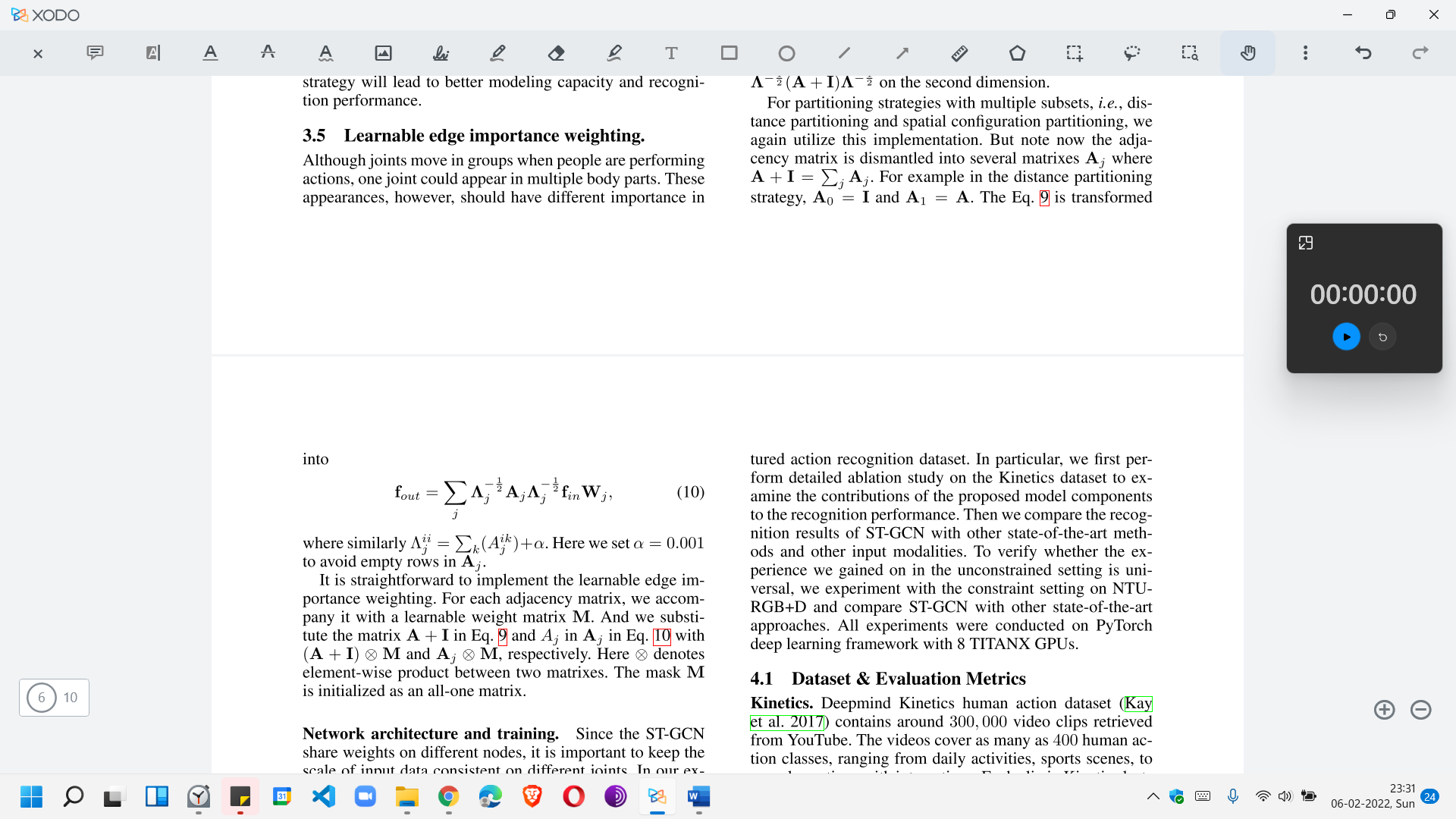
Temporal:

* Neighbourhood: We extend the concept of neighborhood to also include temporally connected joints as B(vti) = {vqj|d(vtj, vti) ≤ K, |q − t| ≤ |Γ/2|}. (Γ is the temporal analogue of temporal Kernel size)
* Sampling function : same as spatial
* Labelling Map : lST(vqj) = lti(vtj) + (q − t + |Γ/2|) ×K ; as the temporal axis is well ordered

Learneable Edge Importance Weighing:

learnable mask M on every layer of spatial temporal graph convolution. The mask will scale the contribution of a node’s feature to its neighbouring nodes based on the learned importance weight of each spatial graph edge in ES (empirically found to increase efficiency of recognition)

Implementation:



Multiple layers of spatial-temporal graph convolution operations will be applied on the input data and generating higher-level feature maps on the graph. It will then be classified by the standard SoftMax classifier to the corresponding action category. We may incorporate Resnet + Random turn off type architectures. The whole model is trained in an end-to-end manner with backpropagation

[0]

Steps:

1. Skeleton Estimation + ST-Graphs construction

*Spatial Perspective*:

*Sampling Strategy*: (D=1)

Spatial = Root as well as distance 1 points connected are sample for Convolutional Filter (spatial edges)

Temporal = union of sampling area from stacked frames (temporal edges

*Partitioning Strategy*: (Spatial Configuration Partitioning based on COG (avg of all joints))

root node (green: l=0) + centripetal group (near COG, blue: l=1) + centrifugal group (far from COG, yellow: l=2)

Reason:

* Unilabelling: Useless
* Distance: D=1 so crude and less useful

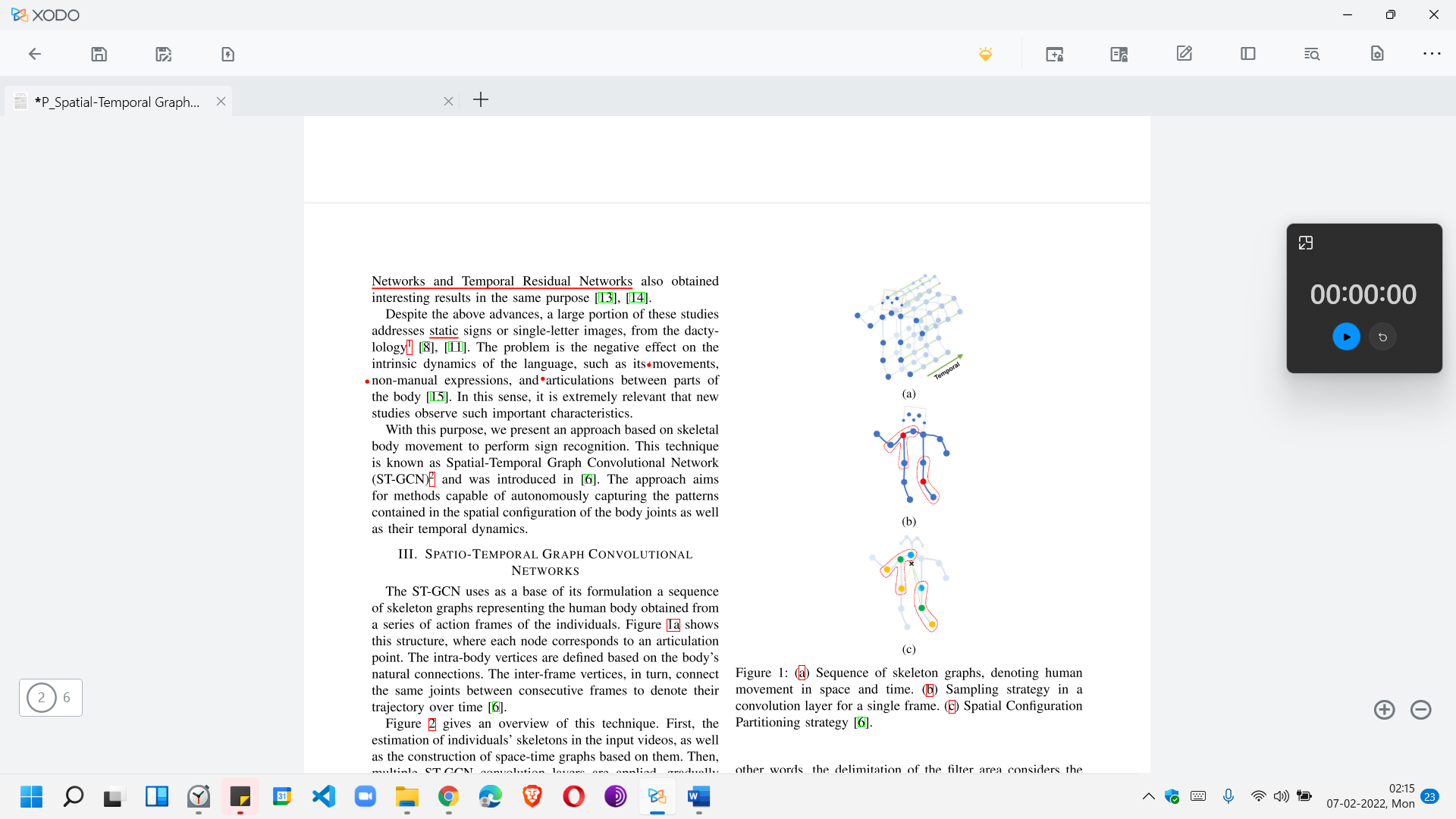
More complex partitions increase efficiency

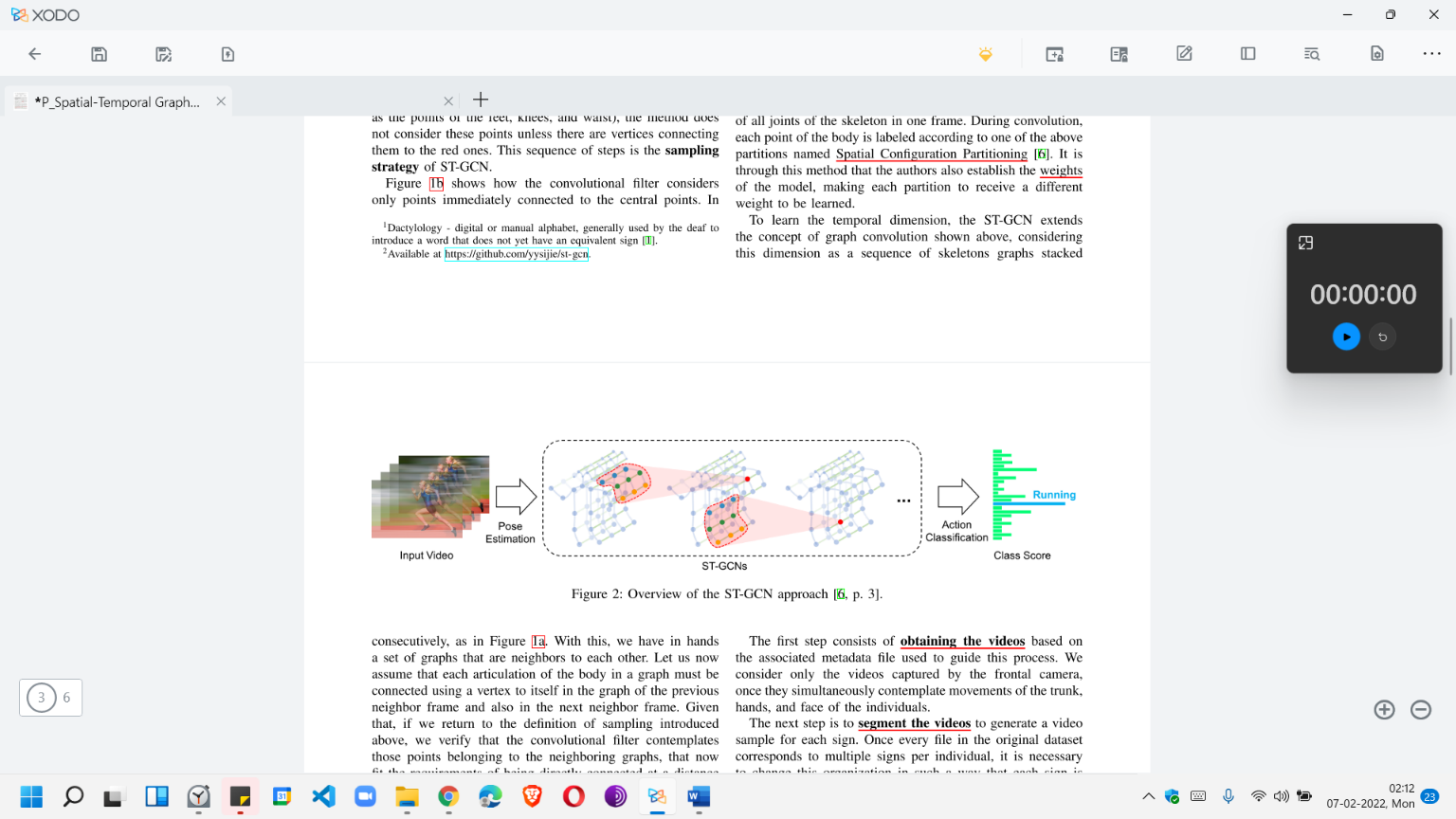
*Suited ST-GCN for SLT*: (keeping architecture same, we change specific adaptations)

(2) Multiple ST-GCN convolution layers are applied, gradually generating higher and higher levels of feature maps for the presented graphs.

We are dealing with convolutions over 2D

1. Classifier identifies the action.





**2. DataSet**

I: ASLLVD samples compatible with the ST-GCN model input.

O: New dataset consisting of the skeletal estimates for all the signs contained therein (ASLLVD-Skeleton)

Steps:

1. Obtaining the videos based on the associated metadata file used to guide this process.

* we consider only the videos captured by the frontal camera, once they simultaneously contemplate movements of the trunk, hands, and face of the individuals.

1. Segment the - each sign is arranged individually with its label.
2. Estimating skeletons – 130 Joint Coordinates via OpenPose library
3. Filtering the key points – 27/130 estimated key points (= 5 for shoulders and arms and 11 for each hand)
4. Train-test split – (80% train + 20% test via SciKit-Learn after shuffling as the set is small (10k approx.))
5. Normalize - repeat frames in case of less frames to make to 63 frames (2s by 30 FPS)

Serialize – translating preloaded normalised samples to physical python files

**Limitations/next steps?**

Limitations:

* Performs poorly with Kinetics train set with mixed videos (with standard configurations)
* Performs worse than HOF and BHOF models

Next Steps:

* Explore other spatial portioning strategies (new partitioning strategy that would allow more emphasis on the subtle traces of hands and fingers to the detriment of the other parts of the body)
* Explore the possibility of temporal partitioning (based on the average duration)
* Sample out larger distances (larger filters)
* Definition of specific weights for these parts would enable the model to learn more about its dynamics – current model does not distinguish the type of learned joint.
* Including the depth information of coordinates may provide to the model more details about the trajectory of these parts, enabling the observation through the three-dimensional plane
* Considering HOF/BHOF as alternatives/ work along with ST-GCN. Also taking into account the recent techniques for SLT (given in the table)
* Considering alternatives to datasets from ASLLVD eg. MSR Gesture3D, Auslan (Australian Sign Language) data set, LTI-Gesture Database, RWTH German Fingerspelling Database, DEVISIGN Chinese Sign Language dataset, Indian Sign Language dataset.

**Where will it be helpful in our project…**

* Sign Language Recognition (ST-GCN clearly outperforms other SOTA skeletal recognition modals + flexible so changes are easy)
* Data – collection of ISL