A Appendix

A.1 Dataset Information

Dataset access and maintenance plan The *MinT* dataset will be provided via the persistent long-term storage service RADAR4KIT (Research Data Repository for KIT), ensuring both uninterrupted and machine readable access. Data published by *RADAR4KIT* is indexed via Metadata following the *Open Archive Initiative* interface which is automatically published to datacite.org and will automatically be referable via a DOI. Data is secured according to *Open Archival Information System* standard ISO 14721:2003 and availability is guaranteed for a minimum of 10 years.

To facilitate the review process and integrate reviewer feedback concerning the data structure (RADAR4KIT data can not be changed easily), we provide an intermediate link for direct download of our data, which will be exchanged with a RADAR4KIT link for the camera ready version.

Currently the dataset can be downloaded under this link (2.2 GB, compressed tar file):
https://s.kit.edu/mint-data

Our code for motion to muscle estimation can be found here:
https://github.com/simplexsigil/motion2muscle.git

License The MinT dataset is build on top of the KIT Whole-Body Human Motion Database, BMLmovi, BMLrub, the EyesJapan dataset and TotalCapture. We make use of AMASS to map from the motions of these original datasets to virtual marker positions in OpenSim.

All of these datasets allow usage of their data for non-commercial scientific research:

- The license of AMASS can be found under https://amass.is.tue.mpg.de/license.html
- The License of BMLmovi and BMLrub can be found under https://www.biomotionlab.ca/movi/
- The KIT Whole-Body Human Motion Database can be used upon citation of the original work as explained here https://download.is.tue.mpg.de/amass/licences/kit.html
- The license for the EyesJapan dataset can be found under http://mocapdata.com/Terms_of_Use.html
- The license for the Total Capture dataset can be found under https://cvssp.org/data/totalcapture/

The Muscles in Time dataset will be published under a CC BY-NC 4.0 license as defined under https://creativecommons.org/licenses/by-nc/4.0/. Researchers making use of this dataset must also agree to the licenses mentioned above which can add additional restrictions depending on the individual sub-dataset.

Our data generation pipeline is licensed under Apache License Version 2.0 as defined under https://apache.org/licenses/LICENSE-2.0.

Code for training our muscle activation estimation networks is licensed under the MIT license as defined under https://opensource.org/license/mit.

Author statement The authors of this work bear the responsibility for publishing the MinT dataset and related code and data.

Data structure The structure of the provided MinT data is intentionally kept simple. All data is saved in CSV files or pandas DataFrames stored in pickle files. In Listing 1 we display how data for an individual sample can be loaded with minimal dependencies (*joblib* and *pandas*). We provide muscle activations in a range of $[0, 1]$, ground reaction forces and effective muscle forces. Data is provided with 50 fps, each dataframe is indexed by fractional timestamps. Columns are named meaningfully, the first 80 muscles belong to the lower body model, the following 322 muscels belong to the upper body model. The first and last 0.14 seconds are cut off since the muscle activation analysis is unstable towards the beginning and end of data. Since the data is generated in chunks of 1.4 seconds and muscle activation analysis can fail to succeed due to various factors, the provided data may contain gaps identified by missing data for certain time ranges.

```
>>> # First download and extract the dataset.
>>> # Example for sample
>>> #'BMLmovi/BMLmovi/Subject_11_F_MoSh/Subject_11_F_10_poses'
>>> import joblib
>>> joblib.load("muscle_activations.pkl")
       LU_addbrev_l    ...    TL_TR4_r   TL_TR5_r
0.14           0.016    ...       0.003      0.061
0.16           0.028    ...       0.005      0.070
0.18           0.033    ...       0.002      0.080
...              ...    ...         ...        ...
3.74           0.024    ...       0.020      0.028
3.76           0.016    ...       0.009      0.004
3.78           0.011    ...       0.003      0.000

[183 rows x 402 columns]

>>> joblib.load("grf.pkl")
       ground_force_right_vx   ...   ground_torque_left_z
0.14                  15.962   ...                    0.0
0.16                  10.596   ...                    0.0
0.18                   3.422   ...                    0.0
...                      ...   ...                    ...
3.72                  20.337   ...                    0.0
3.74                  21.572   ...                    0.0
3.76                  22.546   ...                    0.0

[182 rows x 18 columns]

>>> joblib.load("muscle_forces.pkl")
       LU_addbrev_l   ...   TL_TR4_r   TL_TR5_r
0.14          8.430   ...      0.153     11.652
0.16         15.345   ...      0.283     13.240
0.18         19.127   ...      0.143     15.240
...             ...   ...        ...        ...
3.72         14.437   ...      1.320      3.661
3.74         13.993   ...      1.270      5.330
3.76          9.346   ...      0.577      0.847

[182 rows x 402 columns]
```

Listing 1: Simplified loading of MinT samples with joblib and pandas.

The *musint* package To further facilitate the usage of the MinT dataset, we provide the *musint* package, a Python package that allows data to be loaded into a predefined torch dataset and allows simplified cross-referencing with BABEL dataset labels. Additionally, it includes functionality for sampling a sub-window of the data at a given framerate as well as identifying and handling any gaps in the data. A short example on the musint package usage is displayed in Listing 2.

The *musint* package can be installed via `pip install musint`. Additional insight can be found on the musint github page where we also provide a Jupyter notebook for displaying the data as well as

additional information on muscle subsets:
https://github.com/simplexsigil/MusclesInTime

```
>>> # First download and extract the dataset.
>>> import os
>>> from musint.datasets.mint_dataset import MintDataset

>>> md = MintDataset(os.path.expandvars("$MINT_ROOT"))

>>> md.by_path("TotalCapture/TotalCapture/s1/acting2_poses")
MintData(path_id='s1/acting2', babel_sid=12906, dataset='
    TotalCapture', amass_dur=61.7, num_frames=1114, fps=50.0,
    analysed_dur=22.26, analysed_percentage=0.36, data_path='
    TotalCapture/TotalCapture/s1/acting2_poses', weight=72.1,
    height=169.2, subject='s1', sequence='acting2_poses',
    gender='male', has_gap=False, dtype=object))

>>> md.by_path("TotalCapture/TotalCapture/s1/acting2_poses").
    get_muscle_activations(time_window=(0.3,1.),
    target_frame_count=int(0.7*20.))
       LU_addbrev_l ...  TL_TR4_r  TL_TR5_r
0.30          0.094 ...     0.000     0.020
0.36          0.094 ...     0.003     0.042
0.40          0.091 ...     0.000     0.027
 ...            ... ...       ...       ...
0.90          0.093 ...     0.000     0.008
0.94          0.093 ...     0.000     0.000
1.00          0.094 ...     0.001     0.009

[14 rows x 402 columns]
```

Listing 2: Loading the MinT dataset with the python musint package.

A.2 Additional statistics and information

In Figure 9 we provide additional information on the data analyzed provided with Muscles in Time. Total Capture makes up a small part of the dataset with exceptionally long sequences. The Eyes Japan Dataset provides the largest contribution with 3.2h of analyzed recordings.

In Tables 3 and 4, we list larger muscle groups in the lower and upper body model as well as their function for human motion. Muscle groups or larger muscles can be represented by multiple simulated muscles, e.g. since such muscles are attached to multiple muscle locations or exert forces in varying directions. The *Gluteus Medius* muscle is an example with three simulated activations on each side of the body.

A.3 Design choices and more detailed limitations

The muscle-driven simulation, based on the approach by Falisse *et al*. [18], aims to ensure that muscle and skeletal dynamics align closely with given kinematic data while minimizing muscle effort. This process involves finding a solution within the problem space that satisfies an error tolerance and the number of collocation points, which depend on the dynamics of the kinematic data. Collocation points are used to discretize the continuous kinematic and dynamic equations into a finite set of points, making the optimization problem computationally feasible. To mitigate the risk of nonconvergent or nonmeaningful solutions, we implemented safeguards by restricting the deviation between the kinematic information before and after the optimization problem converges.

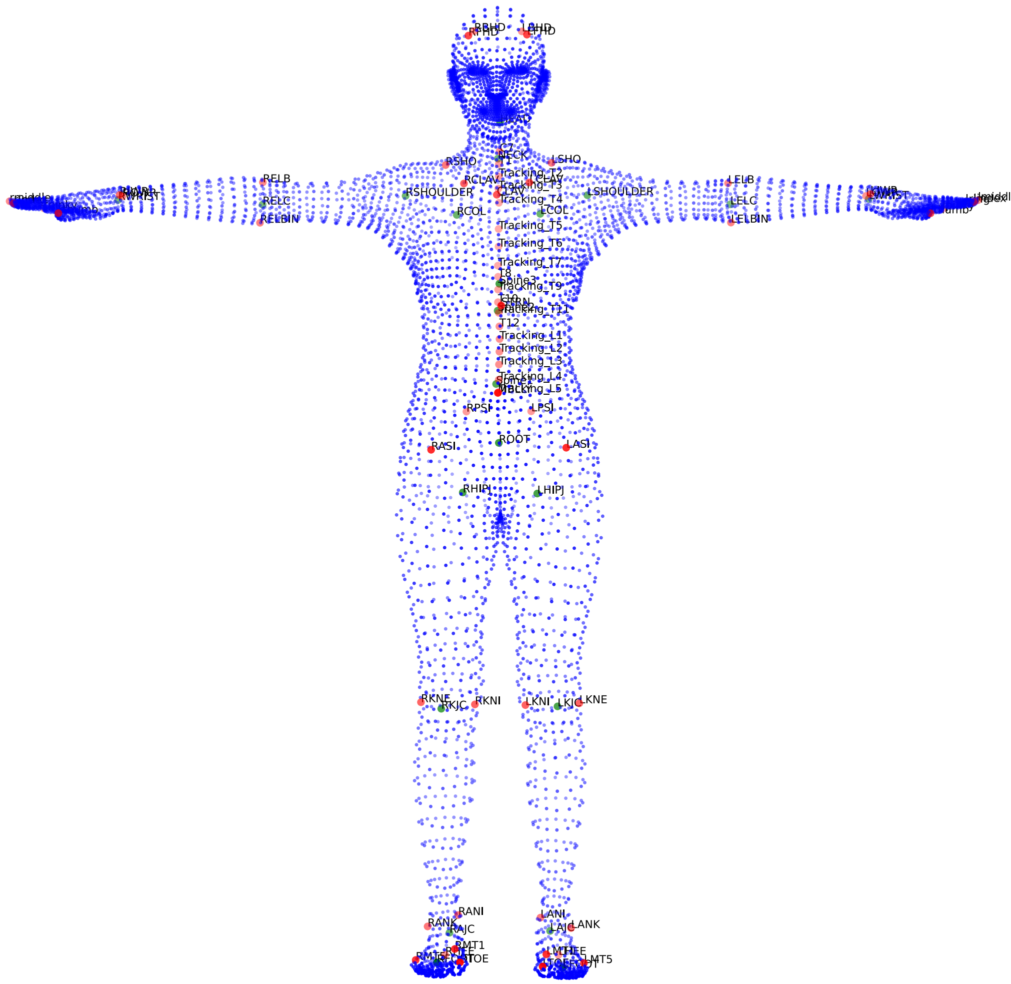

Figure 7: Virtual marker placement for transferring motions to OpenSim, enlarged from Figure 2.

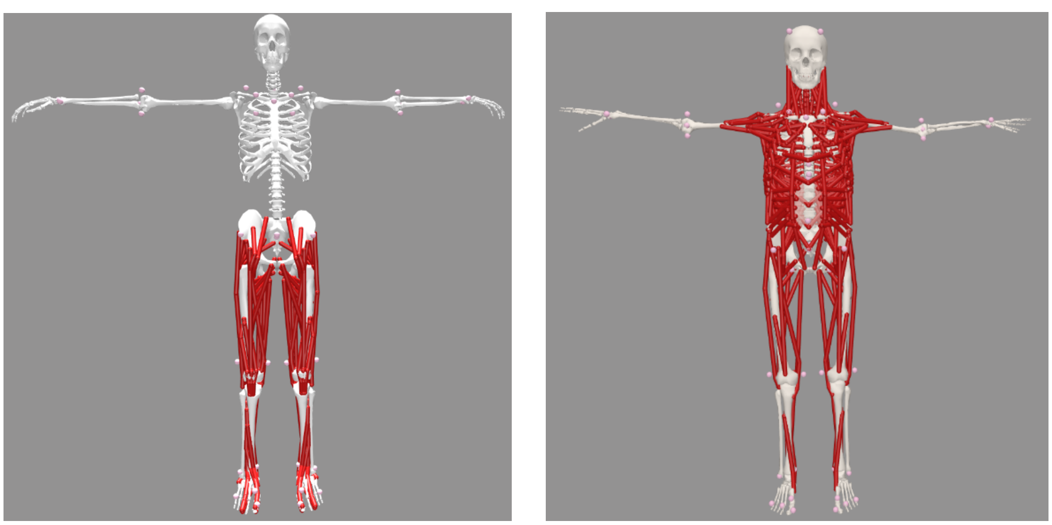

Figure 8: Lower body and upper body model, enlarged from Figure 2.

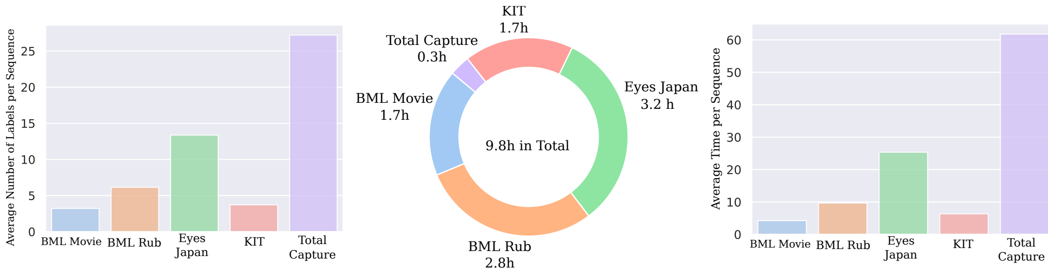

Figure 9: Average number of labels per sequence, composition of sub datasets and average sequence length.

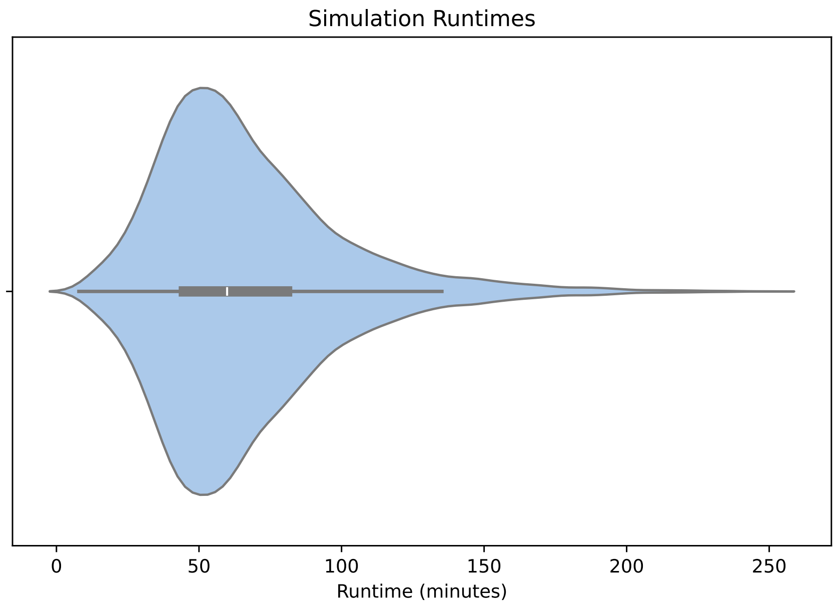

Figure 10: Analysis runtime distribution of the optimal trajectory problem described by Falisse *et al*. [18]. Subset of 10k runs.

Given the computational complexity, we decided to use 50 collocation points per second and an error tolerance of 10^-3. On an Intel Xeon Gold 6230 with 96 GB RAM, processing 6 subsequences of 1.68 seconds (including 0.14 second buffers at start and end) in parallel took approximately a median time of 45 minutes. Figure 10 displays a distribution of sample-wise runtimes in a violin plot. Non-converging samples tend to have higher runtimes and can be found on the long tail on the right. To manage the impact of unsuccessful simulations on the overall runtime, we limited the optimization problem to 2500 iterations and discard a sample if the optimization does not fall within error tolerance after this time. The AMASS sequences were divided into 1.4-second segments to mitigate a nonlinearly increasing runtime associated with longer motion sequences. After simulation, these segments were recombined into the original sequences, with muscle values smoothed at the connection points to ensure seamless transitions.

A challenge arose from minor variable distances between the AMASS body model and the ground, since the contact spheres provided by the OpenCap simulation are susceptible to changes in foot-ground distance. To provide similar foot-ground distances over all AMASS subjects, our pipeline automatically offsets the AMASS model depending on the lowest body marker over the time of the sequence.

Table 3: List of muscle groups modelled in the model by Lai et al. [33], which are analysed in the presented approach, and their functions [74].

Muscle	**Function**
Gluteus Maximus	Extension and rotation of the hip.
Gluteus Medius	Abduction and rotation of the thigh.
Gluteus Minimus	Abduction and rotation of the thigh.
Adductor Brevis	Adduction, flexion, and rotation of the thigh.
Adductor Longus	Adduction and flexion of the thigh.
Adductor Magnus	Adduction, flexion and rotation of the thigh.
Gracilis	Adduction, flexion and rotation of the thigh.
Semitendinosus	Flexion and rotation of the knee, as well as extension of the hip.
Semimembranosus	Flexion and rotation of the knee, as well as extension of the hip.
Tensor Fasciae Latae	Abduction and rotation of the thigh, as well stabilisation of the pelvis.
Piriformis	Rotation and extension of the thigh and abduction of thigh.
Sartorius	Flexion, abduction, and rotation of the hip and flexion of the knee.
Iliacus	Flexion of the hip.
Psoas	Flexion and rotation of the hip.
Rectus Femoris	Flexion of hip and extension of knee.
Biceps Femoris	Flexion of knee and extension of hip.
Medial Gastrocnemius	Flexion of foot and flexion of knee.
Lateral Gastrocnemius	Plantar flexion and knee flexion.
Tibialis Anterior	Dorsiflexion and inversion of the foot.
Vastus	Extension of the knee.
Extensor Digitorum Longus	Extension of toes and dorsiflexion of the foot.
Extensor Hallucis Longus	Extension of the big toe and dorsiflexion of the foot.
Flexor Digitorum Longus	Flexion of toes, as well as plantar flexion and inversion of the foot.
Flexor Hallucis Longus	Flexion of toes, as well as plantar flexion and inversion of the foot.
Peroneus (Fibularis)	Plantar flexion and eversion of the foot.
Soleus	Plantar flexion of the foot.

Mapping AMASS motions to OpenSim models presented difficulties due to the numerous degrees of freedom in the Thoracolumbar model, complicating kinematic analysis. To safeguard the vertebral joints against aberrant movements, we constrained the range of motion for each vertebra, approximating the natural degrees of freedom in the vertebrae joints.

The MinT dataset was restricted to motions involving foot-ground contact only. Motions involving ground contact of other body parts or involving objects were excluded, except for motions that included throwing and lifting, which are particularly relevant for analyzing back muscle activation. In

Table 4: List of muscle groups modelled in the model by Bruno et al. [3], which are analysed in the presented approach, and their functions [74].

Muscle	**Function**
Longissimus	Extension and rotation of the vertebrae.
Iliocostalis	Extension and flexion of the neck.
Semispinalis	Extension and rotation of the vertebrae.
Splenius	Extension and rotation of the vertebrae.
Sternocleidomastoid	Flexion and rotation of the head.
Scalenus	Elevation of ribs and flexion of the neck.
Longus Colli	Flexion of the neck and stabilisation of the cervical spine.
Levator Scapulae	Elevation and adduction of the scapula.
Quadratus Lumborum	Flexion the vertebral column.
Multifidus	Stabilisation cervical vertebrae.
Rectus Abdominis	Flexion of the lumbar spine.
External Oblique	Flexion and rotation of the trunk.
Internal Oblique	Flexion and rotation of the trunk.
Transversus Abdominus	Stabilisation of the trunk.

these cases, we assumed the objects' mass to be negligible, as the AMASS dataset does not provide this information.

A.4 Results for additional muscle subsets

To facilitate comparability to real world recordings as well as to other datasets, we define two muscle subsets of the lower body model, containing either 16 or eight of the most important lower body muscles for human locomotion. The subset `LAI_ARNOLD_LOWER_BODY_16` contains *left gluteus medius 1*, *left adductor magnus ischial part*, *left gluteus maximus 2*, *left iliacus*, *left rectus femoris*, *left biceps femoris long head*, *left gastrocnemius medial head*, *left tibialis anterior*, *right gluteus medius 1*, *right adductor magnus ischial part*, *right gluteus maximus 2*, *right iliacus*, *right rectus femoris*, *right biceps femoris long head*, *right gastrocnemius medial head* and *right tibialis anterior* while the muscle subset `LAI_ARNOLD_LOWER_BODY_8` contains *left gluteus medius 1*, *left gluteus maximus 2*, *left rectus femoris*, *left biceps femoris long head*, *right gluteus medius 1*, *right gluteus maximus 2*, *right rectus femoris* and *right biceps femoris long head*. These subsets are also defined within the musint package.

In Table 5 we list the results of our 16 layer transformer model on these subsets.

Table 5: Human motion-to-muscle activation prediction results for the lower body model.

Motion	All muscles			Lower Body			Subset 16			Subset 8		
	RMSE↓	PCC↑	SMAPE↓	RMSE↓	PCC↑	SMAPE↓	RMSE↓	PCC↑	SMAPE↓	RMSE↓	PCC↑	SMAPE↓
overall	0.036	0.55	95.3	0.048	0.54	45.1	0.066	0.56	47.7	0.060	0.56	45.0
jump	0.052	0.64	100.7	0.051	0.71	52.3	0.059	0.71	55.5	0.056	0.70	54.2
kick	0.046	0.64	102.8	0.053	0.62	54.8	0.068	0.63	57.0	0.067	0.67	57.4
stand	0.033	0.56	97.5	0.046	0.58	45.0	0.062	0.61	47.5	0.052	0.59	43.6
walk	0.026	0.65	90.7	0.044	0.77	42.4	0.060	0.77	43.3	0.057	0.77	43.4
jog	0.033	0.71	99.0	0.046	0.71	51.1	0.063	0.75	51.8	0.062	0.71	52.7
dance	0.041	0.60	109.2	0.057	0.65	58.5	0.073	0.66	59.6	0.072	0.67	59.5

A.5 Training on Muscles in Action

We evaluate the generalizability of MinT by finetuning our 16-layer transformer architecture exclusively on the first and last transformer block and comparing the results with full training from scratch on Muscles in Action [9]. The motions in MIA were obtained with VIBE [32], a 3D pose estimation method performed on 2D images. The resulting motions are very noisy in contrast to the motions in AMASS which are the result of motion capture, inducing a significant domain gap. Table 6 shows our results. We find that limiting our training to the first and last transformer block results in very similar RMSE values compared to full fine-tuning, while PCC and SMAPE clearly displays a small but significant advantage of the full fine-tuning strategy. Still, finetuning the first and last layer only affects some 8% of all trainable weights, and we see this as an indication for the transferability of the knowledge obtained by training on MinT.

Table 6: Human motion-to-muscle activation prediction results on Muscles in Action [9].

Motion	Full Fine-tuning			First and last layer		
	RMSE↓	PCC↑	SMAPE↓	RMSE↓	PCC↑	SMAPE↓
Overall	15.11	0.27	37.0	15.15	0.21	41.6
ElbowPunch	15.66	0.25	43.6	15.48	0.19	48.8
FrontKick	8.49	0.19	34.5	8.20	0.14	41.0
FrontPunch	8.47	0.38	29.8	8.22	0.36	36.3
HighKick	13.09	0.35	37.0	12.94	0.29	39.7
HookPunch	13.18	0.32	37.1	13.28	0.28	44.6
JumpingJack	13.79	0.27	28.5	13.42	0.23	29.5
KneeKick	12.32	0.25	37.3	12.26	0.16	43.0
LegBack	11.70	0.32	37.3	11.91	0.18	44.4
LegCross	13.89	0.17	42.7	13.84	0.11	48.9
RonddeJambe	15.81	0.20	39.5	15.50	0.17	42.6
Running	7.53	0.30	26.3	7.25	0.24	27.4
Shuffle	9.79	0.21	28.0	9.56	0.13	30.5
SideLunges	26.13	0.29	45.9	26.66	0.22	51.7
SlowSkater	20.15	0.26	42.1	20.81	0.19	47.2
Squat	22.68	0.26	44.9	22.76	0.21	48.2

A.6 Additional qualitative examples for MinT

In Figure 6 we listed two qualitative examples to display the muscle activation estimation quality of our best model. In Figures 11 to 17 we display these two test set samples as well es an additional 48 randomly chosen samples from the test set.

A.7 Corrections

In line 266 and 267 we wrote

> In the construction of the dataset, some design choices had to increase simulation robustnees, [...]

while the correct text should be

> In the construction of the dataset, some design choices were made to increase simulation robustness, [...]

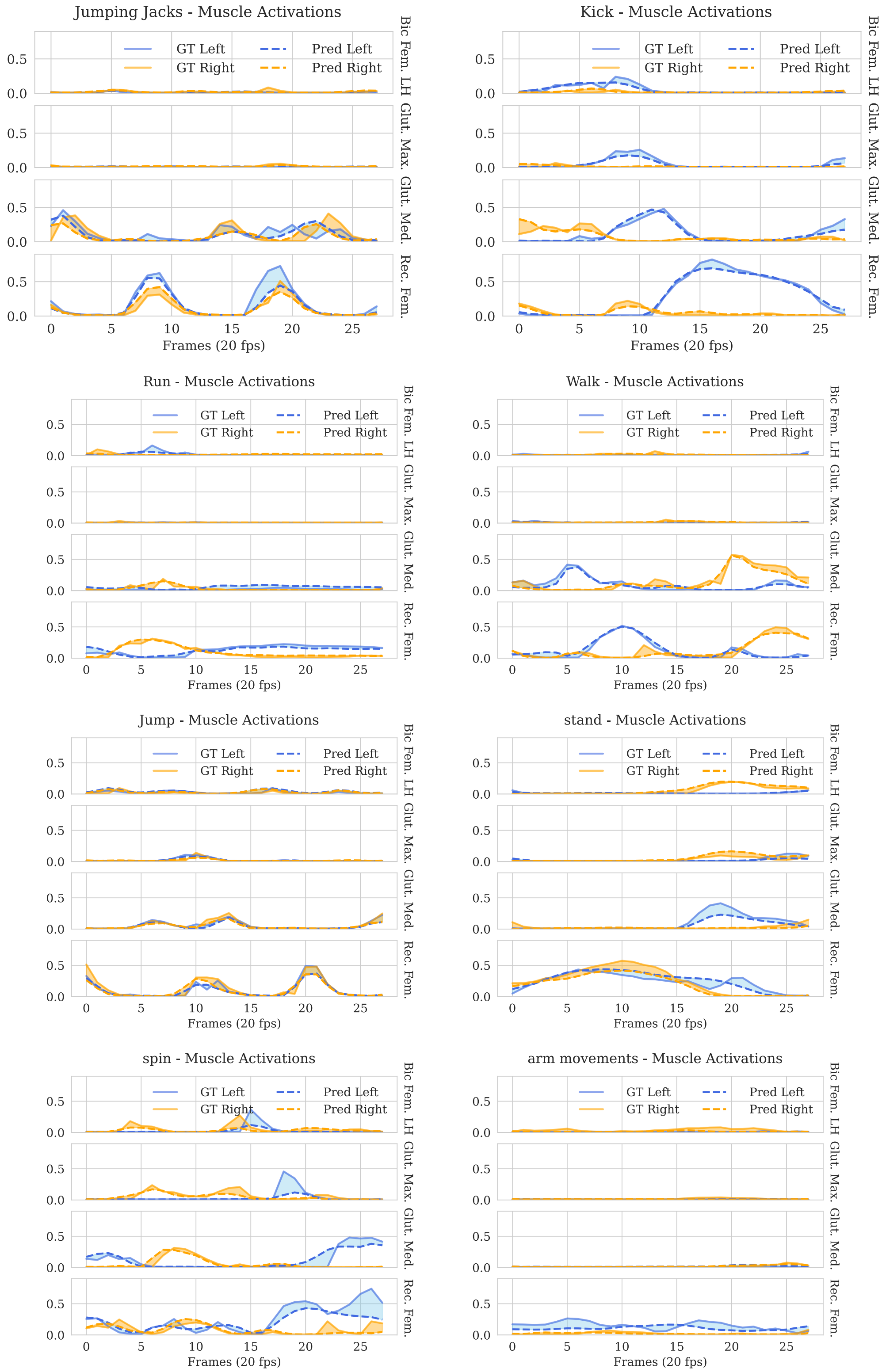

Figure 11: Muscle activation estimation with our 16 layer tranformer model.

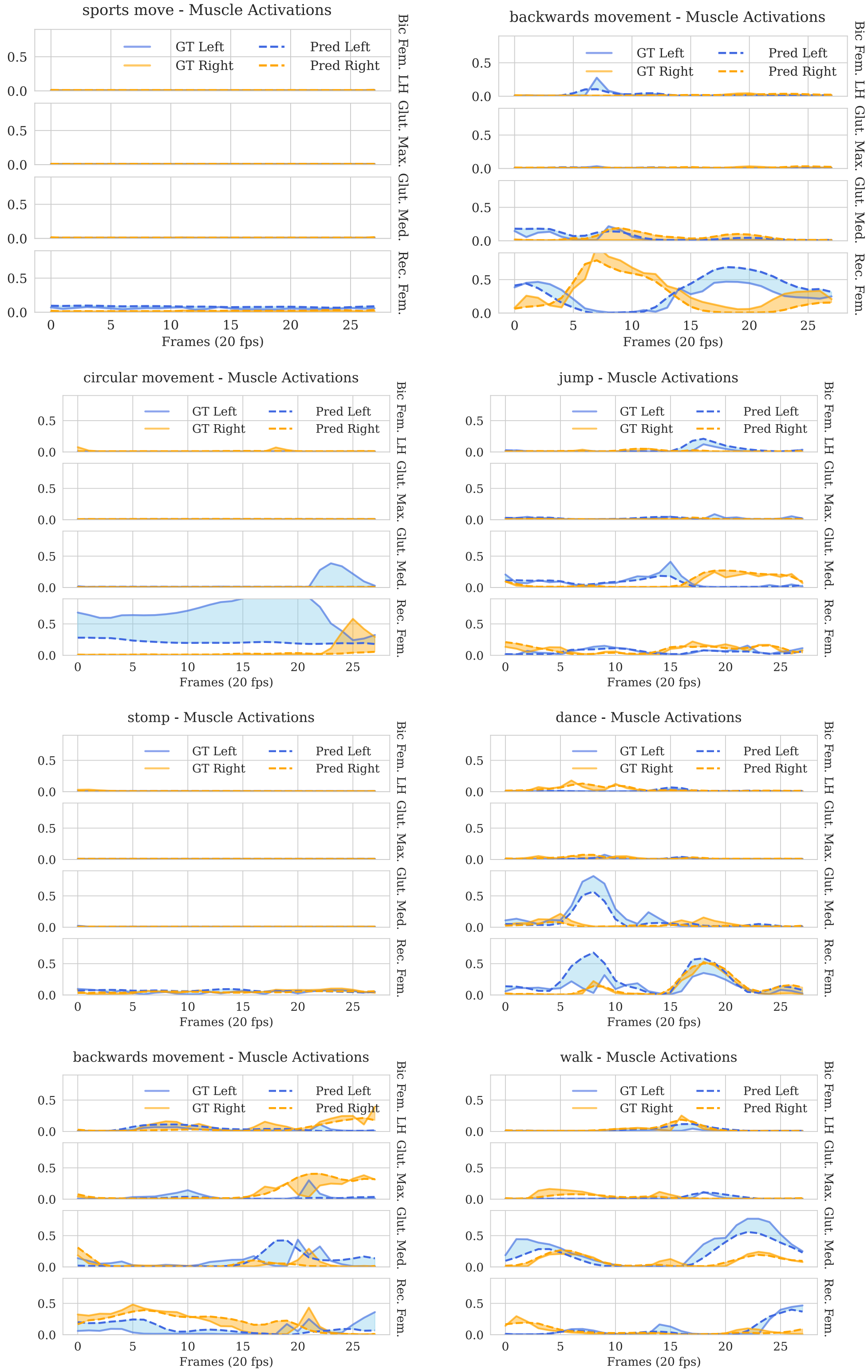

Figure 12: Muscle activation estimation with our 16 layer transformer model.

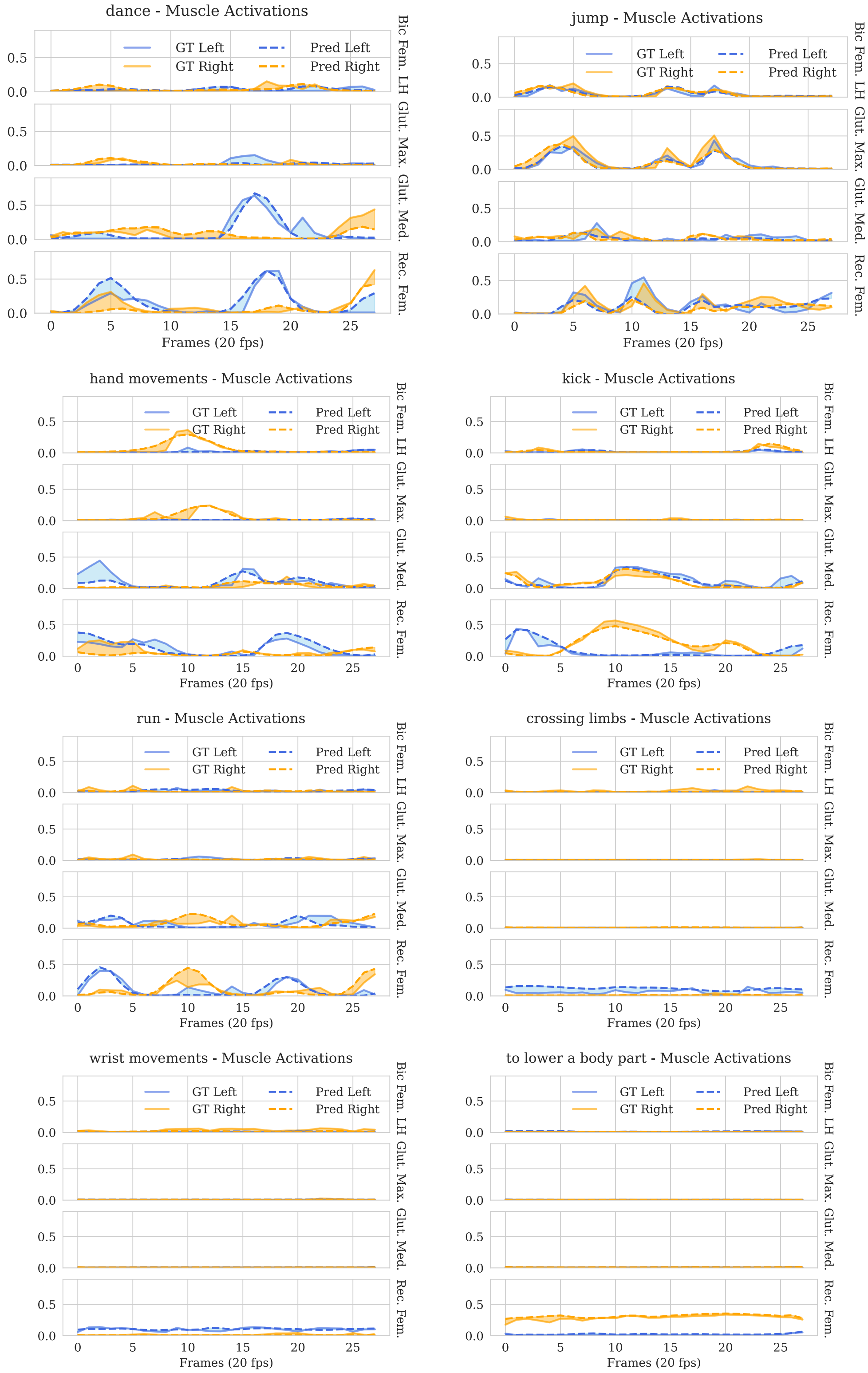

Figure 13: Muscle activation estimation with our 16 layer transformer model.

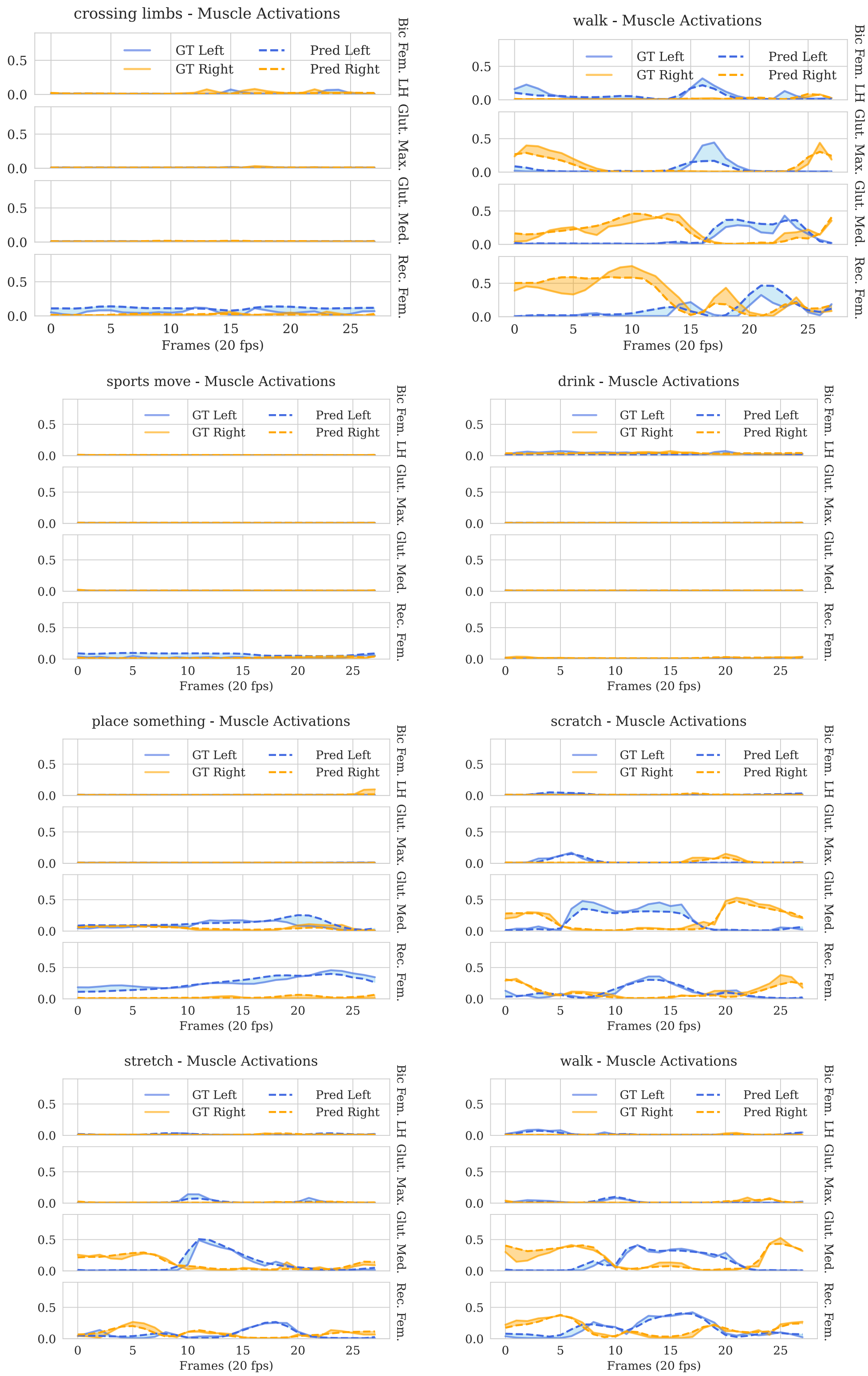

Figure 14: Muscle activation estimation with our 16 layer transformer model.

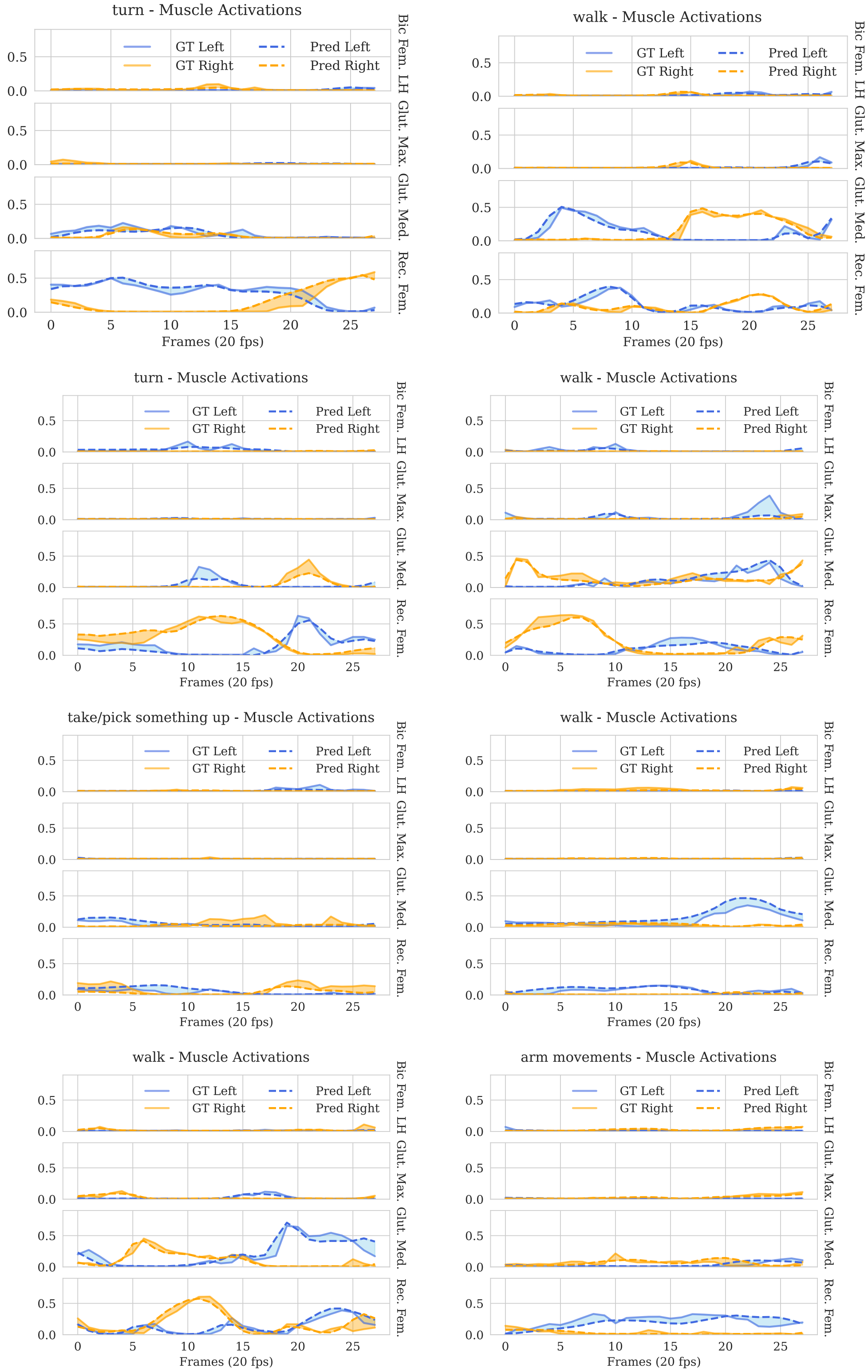

Figure 15: Muscle activation estimation with our 16 layer transformer model.

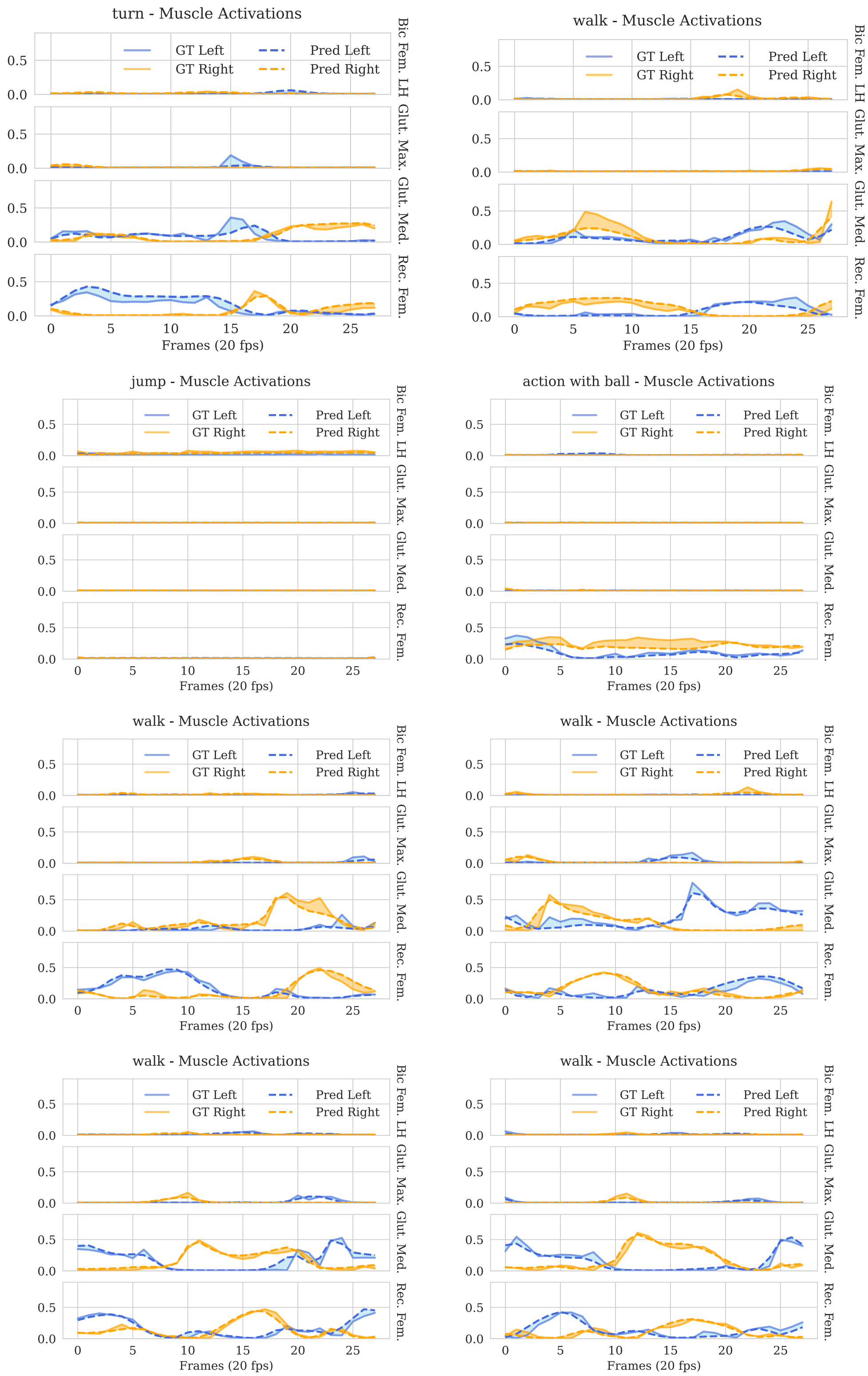

Figure 16: Muscle activation estimation with our 16 layer transformer model.

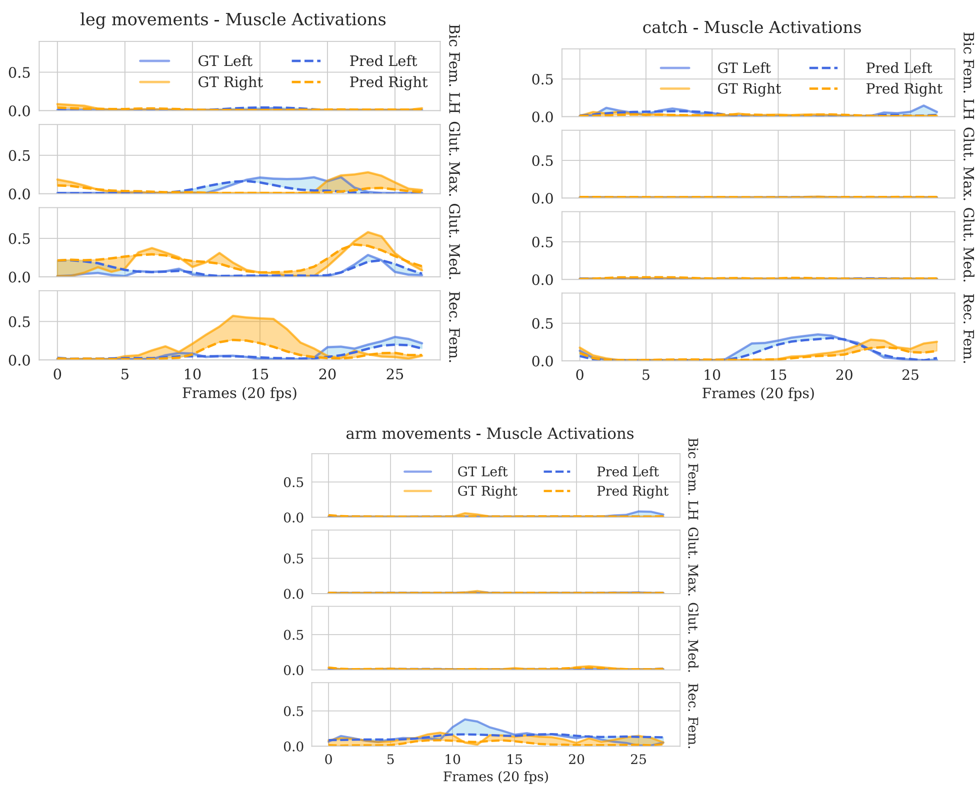

Figure 17: Muscle activation estimation with our 16 layer transformer model.