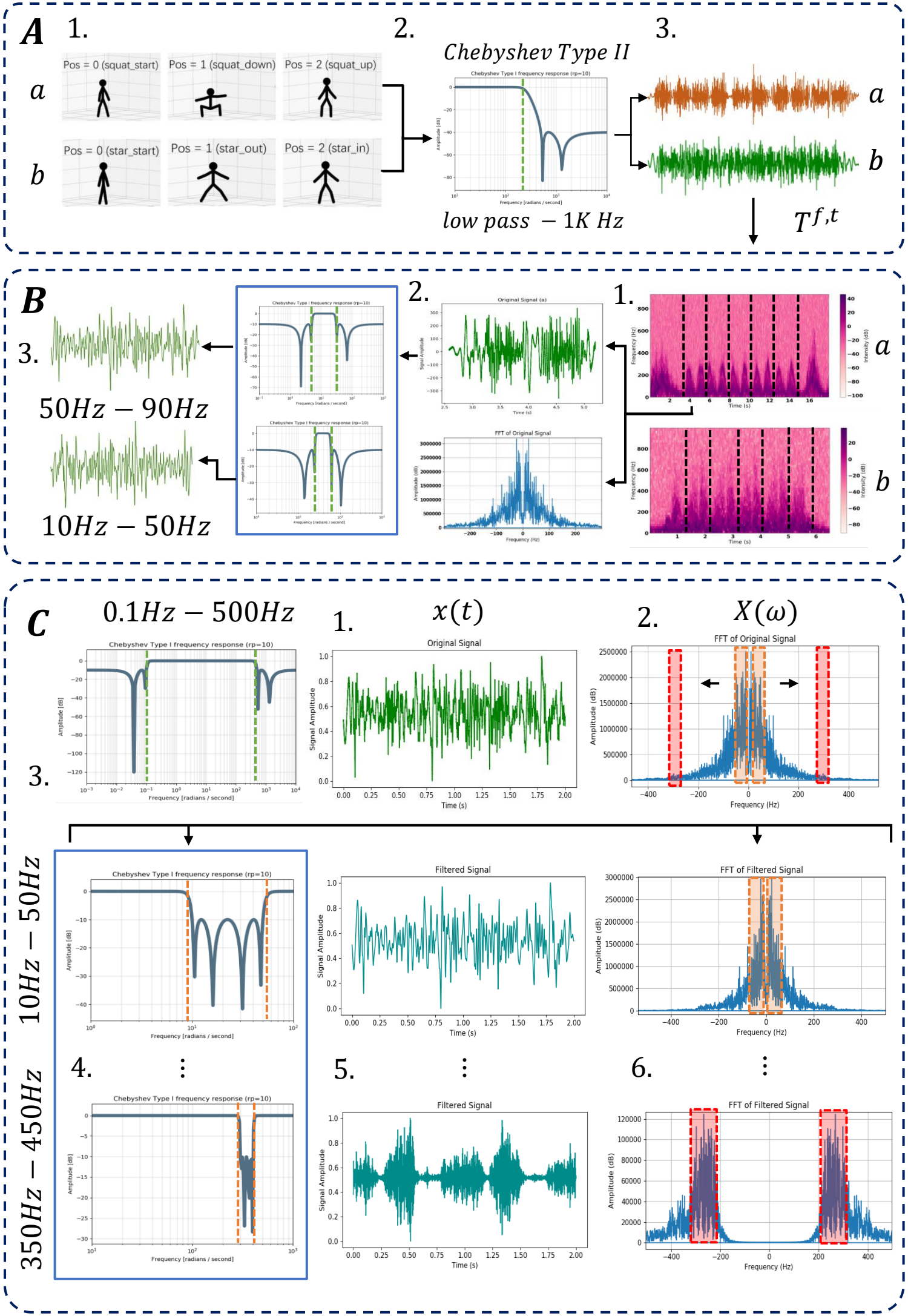
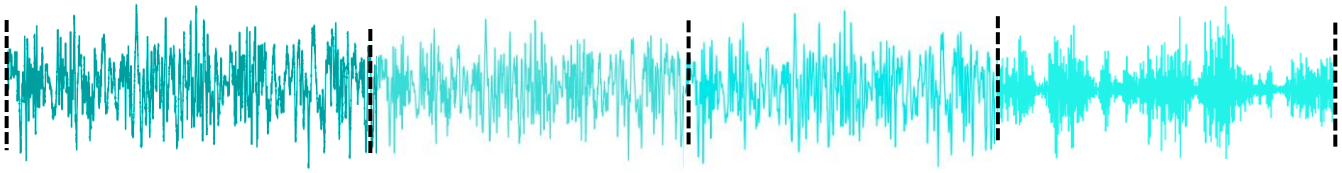

A
1.
2.
3.
a
b
Pos = 0 (squat_start)
Pos = 1 (squat_down)
Pos = 2 (squat_up)
Pos = 0 (star_start)
Pos = 1 (star_out)
Pos = 2 (star_in)
Chebyshev Type II
low pass − 1K Hz
$T^{f,t}$
B
3.
50Hz − 90Hz
10Hz − 50Hz
2.
1.
Original Signal (a)
FFT of Original Signal
C
0.1Hz − 500Hz
1.
$x(t)$
2.
$X(\omega)$
3.
Original Signal
FFT of Original Signal
Filtered Signal
FFT of Filtered Signal
10Hz − 50Hz
350Hz − 450Hz
4.
5.
6.

**A**

2.



$x_0$ $x_1$ … $x_N$

1. $\boldsymbol{X}$

$X_{te}$ (30%) $X_{tr}$ (70%)

3. $f(x, \boldsymbol{W})$

4. $\frac{1}{N}\sum_{i=0}^{N} \hat{\boldsymbol{y}}_i = \hat{\boldsymbol{y}}_{pred}$

5.

| $\hat{\boldsymbol{y}}$ | $\hat{y}_a$ | $\hat{y}_b$ | … |
|---|---|---|---|
| | | | … |
| 'squat' | 'star' | … | |

6. $f(x, \boldsymbol{W})$ $\boldsymbol{t_r}$

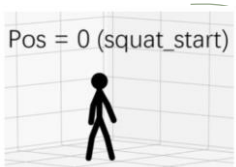
```
ConvNet(
  (layer1): Sequential(
    (0): Conv1d(1, 32, kernel_size=(3,), stride=(1,), padding=(2,))
    (1): ReLU()
    (2): MaxPool1d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (layer2): Sequential(
    (0): Conv1d(32, 32, kernel_size=(3,), stride=(1,), padding=(2,))
    (1): ReLU()
    (2): MaxPool1d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (layer3): Sequential(
    (0): Conv1d(32, 32, kernel_size=(3,), stride=(1,), padding=(2,))
    (1): ReLU()
    (2): MaxPool1d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (layer4): Sequential(
    (0): Conv1d(32, 8, kernel_size=(3,), stride=(1,), padding=(2,))
    (1): ReLU()
    (2): MaxPool1d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (layer5): Sequential(
    (0): Conv1d(8, 8, kernel_size=(3,), stride=(1,), padding=(2,))
    (1): ReLU()
    (2): MaxPool1d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (drop_out): Dropout(p=0.5, inplace=False)
  (fc1): Linear(in_features=26472, out_features=1000, bias=True)
  (fc3): Linear(in_features=1000, out_features=3, bias=True)
)
```

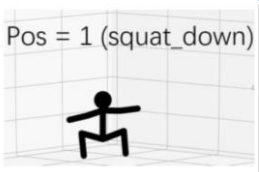
```
out = layer1(x)
out = layer2(out)
out = layer3(out)
out = layer4(out)
out = layer5(out)

out = out.reshape(out.size(0), -1)
out = drop_out(out)

out = fc1(out)
out = fc2(out)
out = fc3(out)

return out
```

```
loss function = CrossEntropyLoss()

Optimiser = Adam(params, lr=0.001, betas=(0.9, 0.999),
                 eps=1e-08, weight_decay=0, amsgrad=False)
```

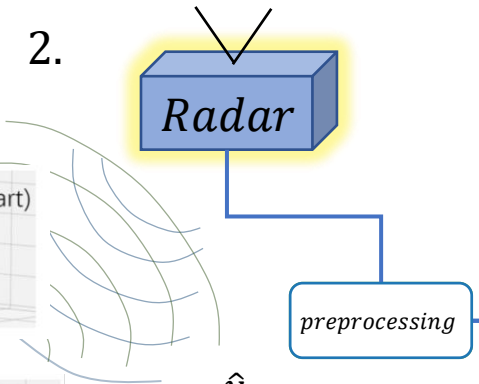
**B**

1.

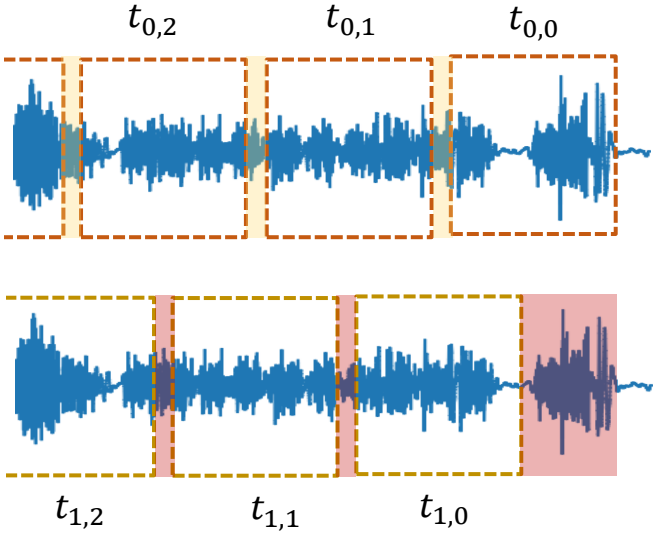



2.




3.




4.