5 Supplementary

Algorithm 1 PyTorch Style Pseudocode for Progressive Knowledge Distillation

Initialize models: S, $\{T_1, T_2, T_3, T_4\}$
Define: N *maxiterations*, C *interval*, *patience*
DataLoader D_{train}, λ, $iter \leftarrow 0$, $T \leftarrow T_1$, $pat \leftarrow 0$, $\mathcal{L}_{KD}^{prev} \leftarrow \infty$
while $iter < N$ and $perf < \tau$ **do**
    $iter \leftarrow iter + 1$
    for batch X, targets Y in D_{train} **do**
        S.train, T.eval
        $P_T \leftarrow T(X_{\text{high res}})$, $P_S \leftarrow S(X_{\text{low res}})$
        $\mathcal{L}_{GT} \leftarrow \text{Loss}(P_S, Y)$
        $\mathcal{L}_{KD} \leftarrow \text{Loss}(S_{bottleneck}, T_{bottleneck}) + \text{Loss}(P_S, softmax(P_T))$
        $\mathcal{L}_{Total} \leftarrow \mathcal{L}_{GT} + \lambda \cdot \mathcal{L}_{KD}$
        Update S using $\mathcal{L}_{Total}$
        if $iter \mod C == 0$ or ($\mathcal{L}_{KD} \geq \mathcal{L}_{KD}^{prev}$ and $pat \geq patience$) **then**
            $T \leftarrow$ Next teacher in $\{T_1, T_2, T_3, T_4\}$
            $pat \leftarrow 0$
        else
            $pat \leftarrow pat + 1$
        end if
        $\mathcal{L}_{KD}^{prev} \leftarrow \mathcal{L}_{KD}$
    end for
end while

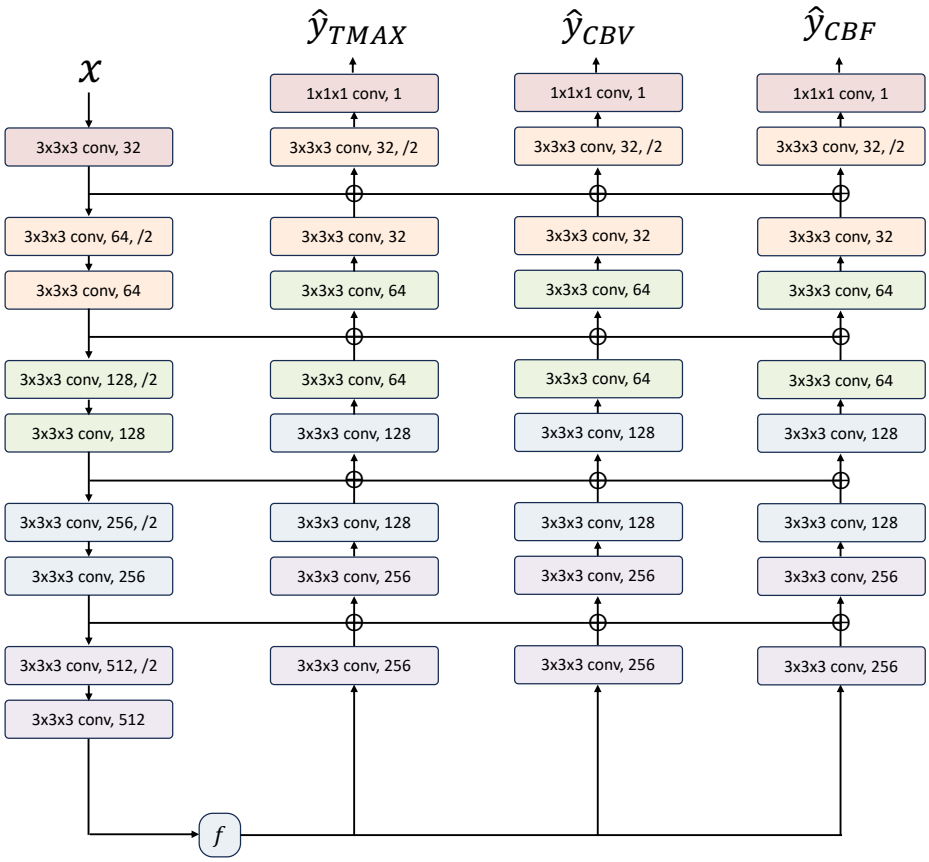

Fig. 3. Detailed architecture of the components in our framework. Intermediate features and bottleneck feature f of the encoder are used as an additional input to each TMAX, CBV, and CBF task specific decoders.