Input: y is a sequence of T characters, V is a subword vocabulary, m is the maximum subword length **Output:** $\log p(y)$ marginalizing out different subword segmentations. 1: $\alpha_0 \leftarrow 0$

2: **for**
$$k = 1$$
 t
3: $\alpha_k \leftarrow 10$

5: return α_T

2: for k=1 to T do

$$\log \sum_{i=k-m}^{k-1} \mathbb{1}[oldsymbol{y}_{i,k} \in V] \exp \left(lpha_i + \log P_{ heta}(oldsymbol{y}_{i,k}|y_1,..,y_n)
ight)$$

3: $\alpha_k \leftarrow \log \sum_{j=k-m}^{k-1} \mathbb{1}[\boldsymbol{y}_{j,k} \in V] \exp \left(\alpha_j + \log P_{\theta}(\boldsymbol{y}_{j,k}|y_1,..,y_j)\right)$

Algorithm 1 Dynamic Programming (DP) for Exact Marginalization

$$\alpha_k \leftarrow \log \sum_{j=k-m} \mathbb{I}[\boldsymbol{y}_{j,k} \in V] \exp \left(\alpha_j + \log P_{\theta}(\boldsymbol{y}_{j,k}|y_1,..,y_j)\right)$$

$$\text{a: end for}$$

4: end for
5: return
$$\alpha_T$$
 \triangleright the marginal probability $\log p(y) = \log \sum_{z \in \mathcal{Z}_n} p(y, z)$