We first prove an intermediate result, which follows trivially from the definition of a representation complexity class.

∀ i, j ∈ [K − 1] : ℛ_T^(i) ⊈ ℛ_T^(j) ⇒ ℛ_T^(i) ∩ ℛ_T^(j) = ∅ (1)

Proof. Let i, j with ℛ_T^(i) ⊈ ℛ_T^(j) and assume that ℛ_T^(i) ∩ ℛ_T^(j) = ∅. For Ỹ ∈ ℛ_T^(i) ∩ ℛ_T^(j), we know that:

R_T(Ỹ) = i ⇔ \min_{\hat{V} \in \mathcal{S}_T(Ỹ)} |\hat{V}| = i,
R_T(Ỹ) = j ⇔ \min_{\hat{V} \in \mathcal{S}_T(Ỹ)} |\hat{V}| = j,

and, hence, is only possible when i = j, which contradicts with the beginning of this proof.

In order to prove Theorem 1, we need to show that the following conditions are met:

1. ∀ i, j ∈ [K − 1] : ℛ_T^(i) ⊈ ℛ_T^(j) ⇒ ℛ_T^(i) ∩ ℛ_T^(j) = ∅
2. \bigcup_{i \in [K−1]} ℛ_T^(i) = \mathcal{P}(Ỹ)

The first condition is met due to Proposition 1. To show that the second condition is met, we need to prove that Ỹ ∈ \bigcup_{i \in [K−1]} ℛ_T^(i) ⇒ Ỹ ∈ \mathcal{P}(Ỹ) ∧ Ỹ ∈ \mathcal{P}(Ỹ) ⇒ Ỹ ∈ \bigcup_{i \in [K−1]} ℛ_T^(i). We start by proving the first part, which follows trivially from the definition of a representation complexity class, as each set that belongs to a given representation complexity class must be element of \mathcal{P}(Ỹ).

To prove the second part, it suffices to show that ∀ Ỹ ∈ \mathcal{P}(Ỹ) : \mathcal{S}_T(Ỹ) ≠ ∅, or in other words, for each element Ỹ in \mathcal{P}(Ỹ) there exists at least one Ṽ ⊂ \mathcal{Y}_T such that:

\bigcup_{v_i \in Ỹ} v_i = Ỹ , \bigcap_{v_i \in Ỹ} v_i = ∅.

Note that each element Ỹ ∈ \mathcal{P}(Ỹ) can be represented by either a node in the hierarchy, the union of sets of leaf nodes in the hierarchy T:

Ỹ = \bigcup_{c_i \in Ỹ} \{c_i\},

or by a union of internal and/or leaf nodes. From this, it follows that \mathcal{S}_T(Ỹ) ≠ ∅ and R_T(Ỹ) = \min_{\hat{V} \in \mathcal{S}_T(Ỹ)} |\hat{V}| = i, where i is lower bounded by one and upper bounded by |Ỹ|. Therefore, given the above, it follows that ∀ Ỹ ∈ \mathcal{P}(Ỹ) : ∃ i ∈ [K − 1] : Ỹ ∈ ℛ_T^(i) which proves the second and last part of this proof.

B EXPERIMENTAL SETUP

We use a MobileNetV2 convolutional neural network [Sandler et al., 2018], pretrained on ImageNet [Deng et al., 2009], to obtain hidden representations for all image datasets. For the bacteria dataset, tf-idf representations are obtained by means of extracting 3-, 4- and 5-grams from the 16S rRNA sequences that were provided in the dataset [Fiannaca et al., 2018]. For the proteins dataset, tf-idf representations are obtained by considering 3-grams only. Furthermore, to comply with literature, the tf-idf representations are concatenated with functional domain encodings, which contain distinct functional and evolutionary information about the protein sequence [Li et al., 2018]. Next, the obtained feature representations for the biological datasets are then passed through a single-layer neural net with 1000 output neurons and a ReLU activation function. We use the categorical cross-entropy loss by means of stochastic gradient descent with momentum, where the learning rate and momentum are set to 1e−5 and 0.99, respectively. For the models without hierarchical factorization, we set the number of epochs to 2 and 20, for the Caltech and other datasets, respectively. For the models with hierarchical factorization, we use 4 and 30, respectively. We train all models end-to-end on a GPU, by using the PyTorch library [Paszke et al., 2017] and infrastructure with the following specifications:

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).
• **CPU:** i7-6800K 3.4 GHz (3.8 GHz Turbo Boost) – 6 cores / 12 threads,
• **GPU:** 2x Nvidia GTX 1080 Ti 11GB + 1x Nvidia Tesla K40c 11GB,
• **RAM:** 64GB DDR4-2666.

Finally, we implemented the RTS and TOP-$k$ algorithms in C++ by using the PyTorch C++ API [Paszke et al., 2017].

**References**


