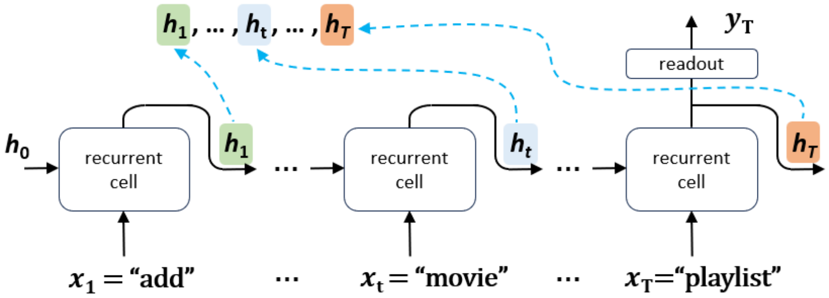

Fig. 2. Sequence of hidden states $\mathbf{h}_1, \ldots, \mathbf{h}_T$ associated with a tokenized input sentence $\mathbf{x}_1, \ldots, \mathbf{x}_T$ (e.g. *"add ... movie ... playlist"*) as it is processed by an RNN. The hidden states $\mathbf{h}_t$ evolve in response to input tokens $\mathbf{x}_t$, representing the progression of the network's internal dynamics. The initial hidden state $\mathbf{h}_0$ serves as the starting point before any tokens are processed. The final hidden state $\mathbf{h}_T$ captures the cumulative information of the input sequence and is used to generate the prediction $\mathbf{y}_T$ via the readout layer. The type of recurrent cell (Vanilla, GRU, LSTM) influences how the network captures sequential dependencies.

this dimensionality [6, 33]. Previous studies suggest that the variance-explained threshold is a robust and empirically validated approach, particularly for classification tasks [3]. This method involves performing Principal Component Analysis (PCA) [25] and determining the number of principal components required to explain a fixed percentage (typically 95%) of the total variance.

Using this approach, we analyzed the state space learned by RNNs trained on the SNIPS dataset. All sentences from the SNIPS test dataset were processed by trained RNNs, and the resulting hidden states were concatenated to form the state space. PCA was then applied to these hidden state points to compute the cumulative variance explained by each principal component. Figure 3 illustrates the results for different RNN architectures with $embed\_dim = 10$ and $hidden\_dim = 20$. The explained cumulative variance is plotted against the number of principal components for the vanilla, LSTM, and GRU cells, represented in blue, yellow, and green, respectively. The variance threshold 95% is indicated by the horizontal dashed red line, and the number of principal components required to exceed this threshold defines the intrinsic dimensionality *id* of the state space. As shown in the figure, the GRU and vanilla RNNs reach a $id = 4$, while the LSTM networks require a slightly higher dimensionality of $id = 5$.

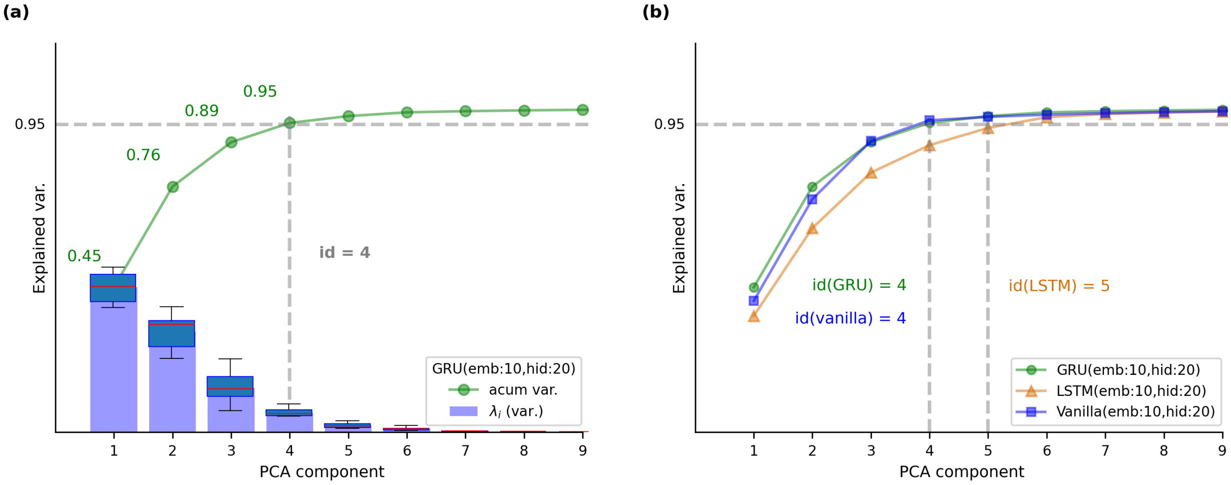

Fig. 3. Variance explained by principal components for the hidden states of RNNs trained on the SNIPS dataset. **(a)** PCA analysis for a GRU-based RNN with embedding size = 10 and hidden layer size = 20. The bars represent the variance explained by individual components, while the green curve shows the cumulative variance. The intrinsic dimensionality (id = 4) is marked where the cumulative variance surpasses the 95% threshold (dashed horizontal line). **(b)** Comparison of intrinsic dimensionalities for GRU, LSTM, and vanilla RNNs with embedding size = 10 and hidden size = 20. The cumulative variance curves show that GRU and vanilla RNNs reach $id = 4$, while LSTM requires $id = 5$ to exceed the threshold.

From related work, it is known that the state space dimensionality of RNNs solving categorical text classification tasks is $N - 1$, where N is the number of classes [3]. For the SNIPS dataset, which contains 7 intent classes, this prediction implies an expected intrinsic dimensionality $id_e = N - 1 = 6$. To investigate whether this hypothesis also holds for intent detection tasks, we performed a comparative analysis of the state space dimensionality and classification accuracy for RNNs with various combinations of embedding layer size and hidden layer size[3]. Figure 4 summarizes the results of this analysis. Each point in the figure represents the average classification accuracy and the median state space dimensionality across training runs with 10 different random seeds. Interestingly, the findings reveal that the dimensionality of the state space is not solely determined by the number of classes, N, as theoretical predictions might suggest. Instead, it is also significantly influenced by the architectural parameters of the network, including the type of recurrent cell, the size of the embedding layer, and the number of neurons in the hidden layer. In particular, the intrinsic dimensionality (id) of the state space is often lower than the theoretically expected dimensionality ($id \leqslant id_e = N - 1$), demonstrating that RNNs can encode task-relevant information in a more compact manner than previously assumed for generic classification tasks.

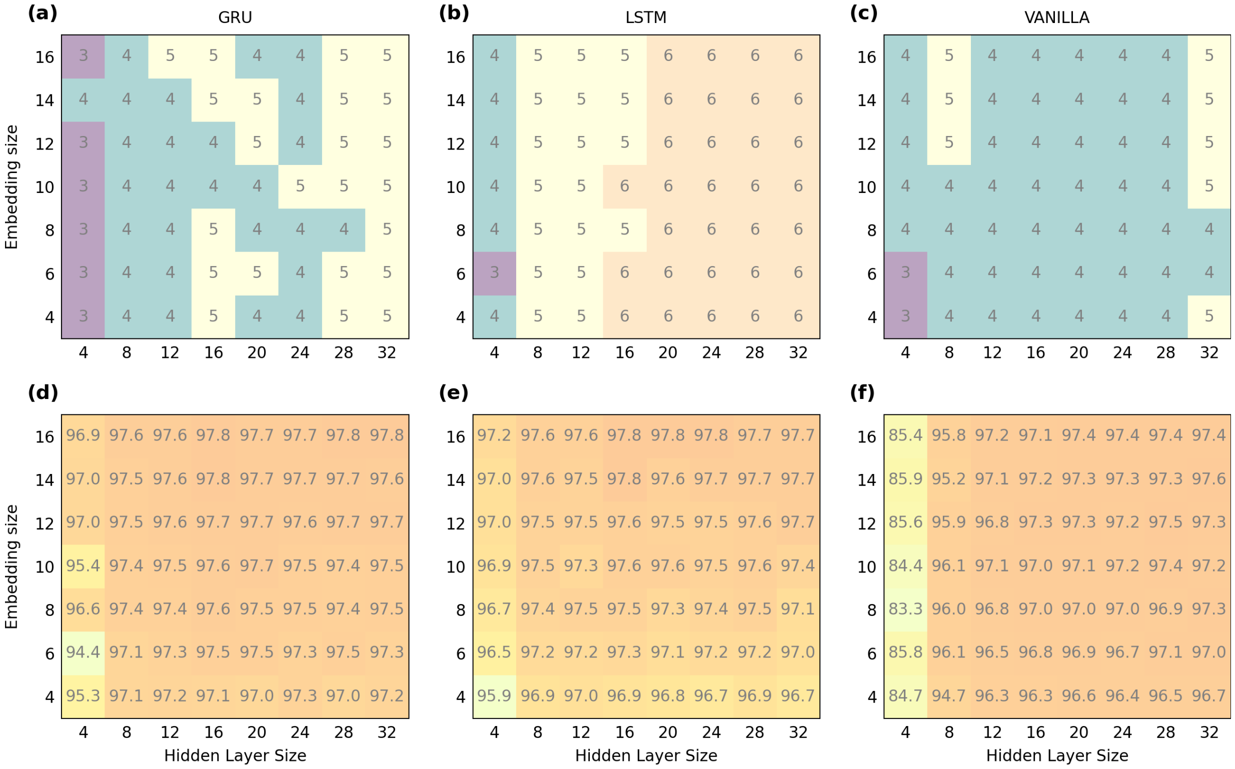

Fig. 4. State space dimensionality and accuracy of RNNs trained on the SNIPS dataset for different combinations of embedding and hidden layer size. The top row shows the intrinsic dimensionality (id) of the state space for **(a)** GRU RNNs, **(b)** LSTM RNNs and **(c)** Vanilla RNNs. The bottom row displays the corresponding classification accuracy for the same architectures: **(d)** GRU, **(e)** LSTM, and **(f)** Vanilla RNNs.

7.2. Intent Detection State Space Projection

In the following section, we analyze the spatial organization of hidden states in trained RNNs, taking advantage of the low intrinsic dimensionality ($id < n\_hidden$) of their state spaces. Our analysis reveals that the trained network

[3] For the sake of simplicity, we denote by *cell_type(emb:x,hid:y)* an RNN with a *cell_type* recurrent unit, *x* neurons in the embedding layer, and a hidden layer of size *y*.

effectively partitions the state space into distinct clusters. Each cluster corresponds to a specific intent, grouping the hidden states of sentences that share the same intent label.

Given a state space, its hidden states can be projected onto a lower-dimensional subspace defined by the top k principal components, as identified in Section 7.1. This projection is achieved by applying a linear transformation $\mathbf{U}$ to each hidden state vector:

$$\mathbf{p}_i = \mathbf{h}_i\mathbf{U} \tag{3}$$

where $\mathbf{h}_i \in \mathbb{R}^n$ is the n-dimensional hidden state, $\mathbf{p}_i \in R^k$ is the projected k-dimensional hidden state, and $\mathbf{U}$ is a $n \times k$ projection matrix whose columns correspond to the top k eigenvectors from the principal component analysis. For visualization purposes, we typically consider projections onto the top-2 and top-3 principal components, which allow the state space to be represented in 2D or 3D, respectively; meanwhile, higher-dimensional projections are useful for dimensionality analysis. Figures 5 (a) and (b) illustrate the 2D and 3D projections, respectively, of the state spaces learned by a GRU(emb:16,hid:16). Each hidden state is colored according to the intent label of the sentence that generated it. The hidden states show a clear clustering pattern, with groups of states corresponding to sentences of the same intent. This clustering suggests that RNNs have successfully learned to organize their state spaces in a way that reflects the semantic distinctions between intents. These visualizations highlight the structured nature of the state space geometry, where each intent is associated with a distinct region in the reduced-dimensional space. The separation between clusters is indicative of the RNNs ability to encode evidence for each intent in an interpretable manner.

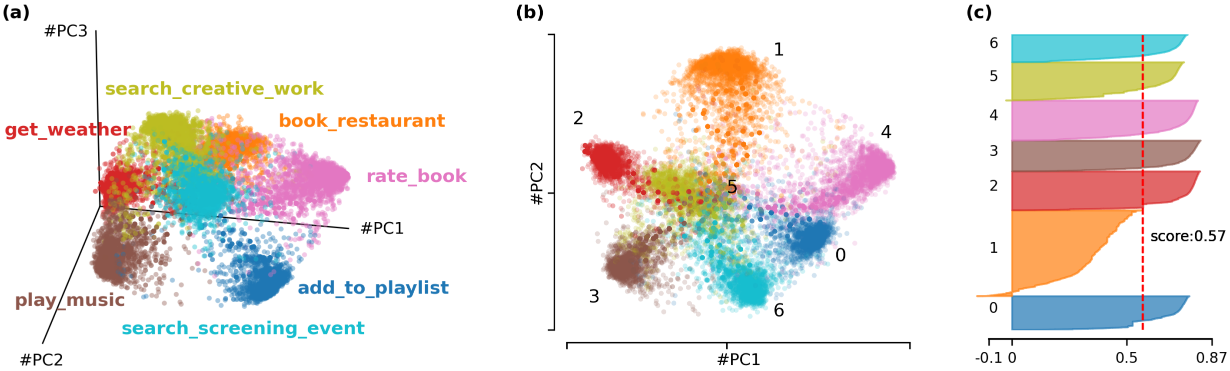

Fig. 5. Top-2 and top-3 PCA projections of the state space of a GRU(emb:16,hid:16) trained on the SNIPS dataset. The hidden states are colored based on the intent label of its corresponding sentence. **(a)** 3D PCA projection. **(b)** 2D PCA projection, highligting intent clusters. **(c)** Silhouette score from a K-means clustering analysis of the state space, showing a score of 0.57 indicating a moderate level of cluster separation.

To numerically verify the presence of clusters in the state space, we applied the classical K-means clustering algorithm [37], configured with 7 clusters (matching the number of intents in the SNIPS dataset), random initialization of centroids and the Euclidean distance metric. The resulting state space partition was evaluated using the silhouette technique [47] which provides a measure of the quality of the clustering. The silhouette coefficient of a point quantifies its separation from other clusters by comparing the average intra-cluster distance (distance to points within the same cluster) to the nearest-cluster distance (distance to points in the closest neighboring cluster). The silhouette coefficients range from [-1, 1]. A value near +1 indicates that the point is well separated from neighboring clusters. Values around 0 indicate points that lie near the decision boundary between two neighboring clusters. Finally, negative values suggest that the point may have been incorrectly assigned to its cluster. The silhouette score is calculated as the mean silhouette coefficient over all points in the dataset. A silhouette score greater than 0.5 is considered indicative of high-quality clusters, where points are well separated and internally cohesive. Figure 5 (c) shows the silhouette scores for a GRU(emb:16,hid:16), where the distances are calculated in the projected state space using only the top-*id* principal components, with *id* corresponding to the intrinsic dimensionality of the state space. The silhouette score exceeds the 0.5 threshold, indicating the quality of the clustering and the clear separation