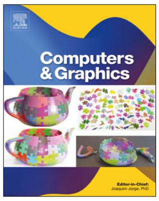
Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag



Technical Section

Concept splatters: Exploration of latent spaces based on human interpretable concepts

Nicolas Grossmann [a], Eduard Gröller [a,b], Manuela Waldner [a,*]

[a] TU Wien, Institute of Visual Computing & Human-Centered Technology, Favoritenstr. 9-11/E193-02, 1040 Vienna, Austria
[b] VRVis Zentrum fur Virtual Reality und Visualisierung Forschungs-GmbH, Donau-City-Straße 11, 1220, Vienna, Austria

ARTICLE INFO

Article history:
Received 24 November 2021
Received in revised form 22 April 2022
Accepted 22 April 2022
Available online 30 April 2022

Keywords:
Latent spaces
Concept spaces
Similarity maps
Visual exploratory analysis

ABSTRACT

Similarity maps show dimensionality-reduced activation vectors of a high number of data points and thereby can help to understand which features a neural network has learned from the data. However, similarity maps have severely limited expressiveness for large datasets with hundreds of thousands of data instances and thousands of labels, such as ImageNet or word2vec. In this work, we present “concept splatters” as a scalable method to interactively explore similarities between data instances as learned by the machine through the lens of human-understandable semantics. Our approach enables interactive exploration of large latent spaces on multiple levels of abstraction. We present a web-based implementation that supports interactive exploration of tens of thousands of word vectors of word2vec and CNN feature vectors of ImageNet. In a qualitative study, users could effectively discover spurious learning strategies of the network, ambiguous labels, and could characterize reasons for potential confusion.

1. Introduction

Similarity maps are often used by the computer vision, machine learning, and visualization community, but also in application domains like biology or geoscience, to better understand what features a network has learned [1]. For this, each data instance is associated with a high-dimensional feature vector defined by the activations of the last hidden layer in the network. These feature vectors are then projected to two dimensions using dimensionality reduction techniques. In the final similarity map scatterplot, data instances are rendered as dots, and those that the network considers similar are rendered in close proximity (see Fig. 1(a)). This encoding can help users to understand implicitly how a model interprets input items [2].

Similarity maps can be found as static images in papers or as one of multiple coordinated views in a visual analytics system [3–5]. They are used to illustrate how the network gradually learns features during the training process [2], at different network layers [3,4], or using different training settings [6]. Furthermore, similarity maps can help to identify mislabeled training samples [7]. Probably the most common usage is to demonstrate whether a network successfully learned to distinguish semantic concepts [8–10] by visualizing if data instances of the same class tend to form separated clusters. Based on the characteristics of the data instances associated with these clusters, it is possible to reason based on which features the network learned this discrimination [11,12].

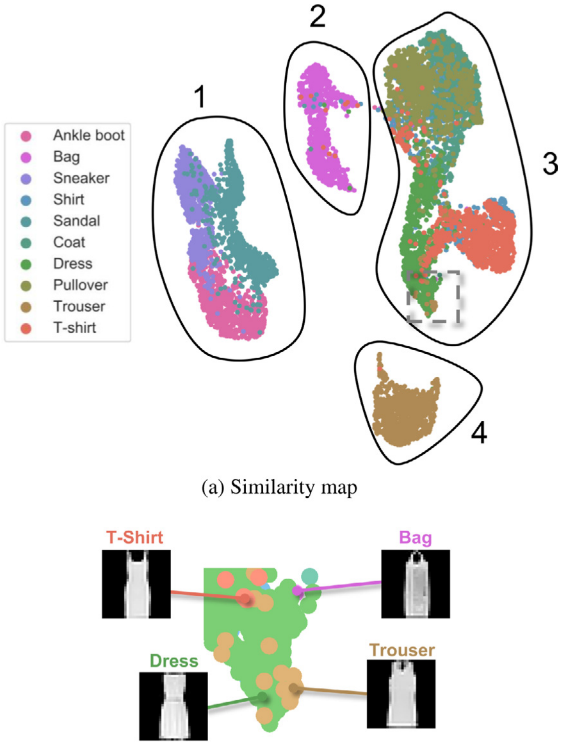
Consider, as an example, the similarity map of a latent space learned by a network from Fashion-MNIST [13] (FMNIST) in Fig. 1. It is clearly visible that the ten fashion categories are not well separated, yet the network tends to cluster the data instances into four distinct groups. Reasoning *why* the categories are not separated well, is not possible in such a view: the axes do not carry any semantic meaning, and it would be required to inspect individual data points to be able to characterize the content of these four clusters. For example in Fig. 1(b), we show images labeled as bags or trousers that are considered similar to dresses (shown in green). From just these four examples, we can already speculate which visual attributes are responsible so that the network considers them similar to dresses: they have a long-stretched visual appearance.

Interactive similarity maps, such as shown in Fig. 1, may still work fairly well with ten ground truth labels. However, more than twelve colors are hard to discriminate [14]. It is therefore clear that classic similarity maps are no longer sufficient when analyzing what a network has learned from a large and more complex dataset, such as ImageNet [15] with 1000 different classes and dozens to hundreds of images associated with each class or large word embeddings, where every word represents its own class.

The goal of this work is to scale up similarity maps so that users can explore if the similarity of a large-scale dataset learned

[*] Corresponding author.
E-mail address: waldner@cg.tuwien.ac.at (M. Waldner).

https://doi.org/10.1016/j.cag.2022.04.013



(a) Similarity map

(b) Inset with Outliers

Fig. 1. Similarity map of a latent space created by a two-layer network trained on FMNIST: the ten ground truth classes are spread over four clusters (a). Few examples are sufficient to grasp their similarity to the class "dress" (b).

by a network (i.e., visual similarity or linguistic contexts) correlates with an expected pre-defined semantic categorization with hundreds to thousands of classes. To this end, we contribute annotated concept splatters as novel foundation for scalable visual exploration of large latent spaces based on human interpretable concepts. The visual encoding principle of concept splatters is based on *prototype theory* [16], which is a theory of cognitive hierarchical categorization, where exemplars serve as representatives of categories. The concept splatters technique has the following novel aspects:

- a multi-scale visualization approach based on a novel combination of hierarchical aggregation in a human interpretable concept space and an adjustable density visualization in machine-learned latent space, which enables new zoom & filter interaction techniques to qualitatively assess what the network has learned on multiple levels of abstraction,
- illustration of agreements and disagreements between concept and latent space through automatically generated insets annotating the concept splatters, and
- new selection-based interaction techniques supporting multi-scale exploration of latent spaces consisting of tens of thousands of samples and labels in multiple coordinated views with interactive frame rates on the web.

We provide an online[1] latent space exploration interface to inspect concepts in large, widely used word embeddings (word2vec [17]) and CNN feature spaces (using ImageNet [15]) with real-time performance. Using this online interface, users of a qualitative study discovered several insightful characteristics of the neural network, such as the presence of spurious learning strategies, as well as potential issues of the inspected dataset, such as ambiguous labels.

We first discuss the state-of-the-art of scalable similarity maps (Section 2) followed by a description of the theoretical background based on prototype theory [16] and the hypotheses derived therefrom guiding the visual encoding principle (Section 3). We then present the visual encoding of concept splatters and their annotations along with a first quantitative validation of the hypotheses in Section 4. Section 5 explains the new zoom & filter and selection interaction techniques, and Section 6 describes the web-based implementation. Finally, we present the data for our use cases in Section 7 and the results from a qualitative thinking-aloud study in Section 8.

2. Related work

Visual exploration of latent spaces can happen in isolation [18–20], but can also be a part of a deeper inspection of how neural networks behave [3–5]. In the majority of cases, visual exploration of latent spaces is at least partially enabled by classic **similarity maps**, i.e., dimensionality-reduced scatterplots of high-dimensional activation vectors of a large dataset. Latent space similarity maps reveal information about sample numerosity and separability of classes. To qualitatively characterize the features a network has learned, users have to manually select map regions [9,19] or individual data instances [4,18] for closer inspection. Such a manual inspection approach does not scale well with increasing numbers of data instances.

A relatively simple approach to create scalable scatterplots is to use a **sampling** strategy [21]. This approach has been used for the latent space similarity map in the network analytics system ActiVis [3], where users are asked to only select a subset of the data for inspection in the latent space scatterplot. The problem of sampling is that it always leads to potential loss of outliers or fine-grained patterns [21].

A simple scalable approach to visualize classed similarity maps is to aggregate all instances per class to a **class prototype** and only draw one dot per class on the class' average position [5,22]. This effectively increases the scalability in terms of the number of instances and reveals classes that are considered very similar, on average. However, due to averaging, intra-class variability is not revealed.

For image-based latent spaces, a popular technique is to assume a **grid** over the whole projected 2D space and show the nearest neighbor image to the center of each grid cell [23]. A similar approach is used for activation atlases, which show feature visualizations of the average activations per grid cell [24]. Although this creates an aesthetic, space efficient overview of the latent space, creating such a grid of images can be considered as a form of sampling strategy. Thus, rare categories and subtle variations may get lost if they are not explicitly detected and visualized [25].

Another approach to ensure scalability is to apply **clustering** in the latent space. Combinations of dimensionality reduction and clustering are very common, not only with respect to neural network analysis [26,27]. Clusters computed from the projected data instances in latent space generate well-defined, non-overlapping regions in similarity maps and can therefore facilitate labeling with word clouds, embedded charts, or images [28]. For each cluster, one representative image [29] or multiple representative words [30] can be shown. Rathore et al. [31] construct a graph out of cluster nodes, where edges connect overlapping clusters, to properly reflect the latent space topology. Ventocilla et al. [32] use a variant of the GNG clustering algorithm [33] to visualize representative data instances and their connections. Clustering

[1] https://kontor.cg.tuwien.ac.at/ConceptSplatters/

from the instances in latent space alone reveals groups of instances the network considers to be similar. However, instances within a latent space cluster may not necessarily be considered as semantically similar by a human observer. Conversely, groups of semantically similar data instances may lead to multiple clusters in latent space. Concept splatters therefore explicitly visualize how groups of instances with high semantic similarity are distributed in the network's latent space. In other words, we visualize if the user's expected categorization is reflected by what the machine has learned.

A general approach to avoid overplotting in scatterplots is to show **densities** instead of individual dots — either as discrete rectangular or hexagonal bins [34], as scalar fields derived from kernel density estimations [35], or bounded density regions for classed scatterplots [36]. In the context of visual latent space exploration, density-based similarity maps have been used to observe the sample distribution of a single user-selected label [37]. In contrast, we want to visualize the distribution of hundreds to thousands of data instances that can be associated with thousands of concepts.

In interactive systems, the approaches listed above can be combined with **hierarchical** aggregation. Using zooming or other types of drill-down interaction techniques, users can interactively reveal more and more details [38]. For instance, zooming into a grid-based activation atlas increasingly reveals further activation labels (i.e., labels that are supported most in one grid cell of activation vectors) [24]. Hierarchical stochastic neighbor embedding (HSNE) [9] iteratively aggregates high-dimensional neighboring sample points into sets of landmarks, and for each layer of landmarks, a t-SNE projection is computed. The main motivation of hierarchical dimensionality reduction techniques stems from the computational demand, but they also implicitly support overview+detail [9] or focus+context [39] exploration of very large, high-dimensional datasets. Similarly to clustering approaches in latent space, landmarks in latent space represent groups that the machine considers to be similar, which does not necessarily correspond to the user's expected categorization.

In contrast of hierarchical aggregation in data space, El Assady et al. [40] proposed a hierarchical aggregation in data *and* a user-generated **concept space** to explore and refine topic models in the context of text analysis. In contrast, concept splatters combine hierarchical aggregation in an explanatory concept space and combine it with a data-driven density visualization in the similarity map. We discuss and demonstrate how this new visual encoding can reveal insightful visual patterns and lead to an illustrative selection of data instances for visual annotation.

3. Theoretical background

Prototype theory states that humans tend to group the stimuli they receive into *basic categories* [16]. A *category* thereby is a named group of objects that are considered to be equivalent, and a *taxonomy* is a hierarchical categorization. A basic category is the *level of abstraction* an observer chooses from a taxonomy so that objects are grouped into a cognitively usable number of categories that can be clearly differentiated from each other. When training a neural network, the expectation is that the network learns a categorization that corresponds to a human interpretable taxonomy. For example, if we train a network to differentiate categories of fashion items on images, we expect it to consider images that contain shoes to be more similar to each other than images of bags. Only if we look closer into shoes, we expect that the network categorizes shoes into sub-categories, such as sandals and sneakers.

Depending on the dataset used to diagnose the network, useful taxonomies can be biological taxonomies, categorizations of objects, such as fashion items, furniture, or vehicles, or topic classifications, such as hierarchical keyword classifications. The manually curated WordNet [41] captures many of such taxonomies into a large lexical database and is also used, for example, to classify images in the large-scale ImageNet dataset [15]. We refer to such a taxonomy of human interpretable concepts as *concept space*.

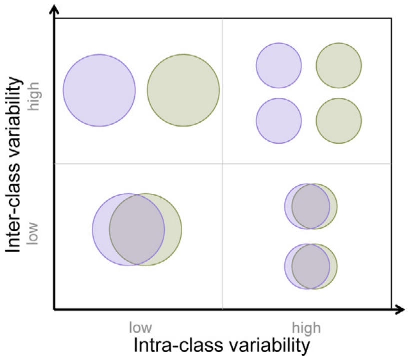


Fig. 2. Patterns observable through visual topological analysis of two basic concept space categories (represented by their color) in latent space.

We postulate that the explicitly or implicitly learned basic categories of a network for a set of data instances can be inferred from its latent space. A latent space is a high-dimensional vector space in which each data item is represented as a single vector. We focus on two types of latent spaces in this work: (1) Word embeddings are created through the analysis of word co-occurrences [17]. They are based on the distributional hypothesis, stating that words in similar contexts have similar meanings. Thus, it is expected that word vectors that cluster together have similar hypernym–hyponym relations [42]. (2) Feature vectors are extracted from the activations of convolutional neural networks (CNNs). Their induced latent space is created through the successive use of convolutions leading to the extraction of higher-level features the further an image progresses through the network. The extracted features of the last layer serve as the basis of the classification. Hereby, images with similar contents are expected to have similar activation vectors [2].

High-dimensional latent space vectors can be visualized in 2D using a dimensionality reduction technique like t-SNE [43] or UMAP [44] to create similarity maps. Dense regions in similarity maps then can be considered to be the basic categories the network has learned. Visual topological analysis of dense class regions allows for a qualitative inspection of (1) the *inter-class variability*, (2) the *intra-class variability*, or (3) *rare categories* of data instances. If well-separated dense regions correspond to the basic concepts of the concept space (i.e., the inter-class variability is high, and the intra-class variability is low, as shown in Fig. 2 top left), this indicates that the network has learned basic concepts as expected from the concept space. In Fig. 1(a), for instance, "bag" and "trousers" have a low intra-class variability so that their data instances form tight clusters. On the other hand, overlapping concept space categories may indicate a low inter-class variability (Fig. 2 bottom) and that the machine cannot properly distinguish them. For example, in Cluster 3 of Fig. 1, multiple fashion categories overlap, such as T-shirt, shirt, and pullover. High intra-class variability (Fig. 2 right) could be an indication that the network has learned to categorize images based on spurious correlations, such as watermarks in images [45], the context in which an object is depicted in images, as well as ambiguous or incorrect labels [7]. Fig. 2 summarizes possible patterns that can emerge when mapping basic concept space categories onto dense regions

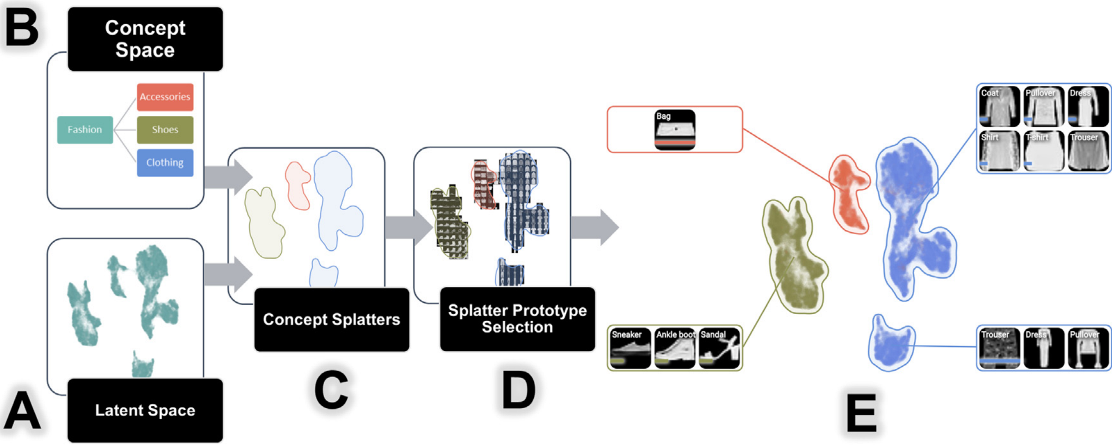


Fig. 3. Overview of the concept splatters generation process: From the distribution of data samples in the 2D projected latent space (A) and a pre-defined hierarchical concept space (B), we find dense region of samples associated with basic categories (the *concept splatters*) (C). From the samples contained within these dense regions (D), we select representative prototypes, which are then used to annotate the concept splatters (E).

of data instances in the latent space visualization. In summary, we hypothesize that *visualizing the distribution of basic categories of the concept space in the latent space can reveal if a machine has successfully learned the expected categorization expressed in the concept space* (**H1**).

Rosch et al. [16] explain that basic categories can be represented by *prototypes*, i.e., the most characteristic instances of a category, which share the most of their attributes with other members of the category. Indeed, to communicate their interpretations visually, authors of machine learning papers often manually augment figures of similarity maps with hand-picked samples [2,10] or carefully selected detail insets [8,11,12]. Therefore, we hypothesize that *few prototype data samples of dense regions associated with basic concept space categories in the latent space are sufficient to characterize what the network has learned on a given level of abstraction* (**H2**).

4. Visual encoding

In this section, we describe how we perform the mapping of the basic categories of the concept space into a similarity map visualization of the latent space. This linkage is the foundation for our interaction design presented in Section 5.

Let S be the set of data samples whose activations by the network should be visualized in the latent space. Each sample has a set of one or more associated labels $L(s)$, a 2D coordinate in the dimensionality reduced latent space $\mathbf{x}(s)$ (see Fig. 3A), and a representation of itself, such as an image thumbnail. Although our approach is agnostic to the underlying dimensionality-reduction method, we specifically chose UMAP to obtain $\mathbf{x}(s)$. It preserves the global structure well, enabling the exploration of overarching patterns in the data [44,46], and is significantly faster to compute than t-SNE [47].

Let L be the set of unique labels associated with all samples. These labels can be ground truth class labels of images or the words themselves in word embeddings. Let C be the set of human interpretable concepts (the *concept space*), which are organized as a rooted tree, where each child node represents a separate subset of all human interpretable concepts, which can be further subdivided (Fig. 3B). Each label $l \in L$ now needs to be mapped to one or multiple concepts of C. For example, an image tagged with the text label "dog" can be associated with seven synsets in WordNet, from a domestic pet to an informal term for a man. To link all samples $s \in S$ with concepts, we therefore find $C(s) = \{c | c \in C, c \sim l \in L(s)\}$, which contains all concepts that match the sample's labels.

We are linking the latent space (i.e., what the machine considers to be similar) and the concept space (i.e., what the users consider to be similar) through *concept splatters*. Concept splatters are continuous regions in the projected latent space and enclose samples that are similar with respect to both, the concept space and the latent space (Fig. 3C).

4.1. Mapping concepts to latent space

The level of abstraction in the concept space is set by selecting a subset of concepts rooted at the selected concept C_r in the concept tree. We create concept splatters for all n proper subsets of $C_r = \{C_1, C_2, \ldots, C_n\}$, i.e., the basic categories of the chosen level of abstraction in the concept space. For the ith subconcept of C_r, we then find all samples $S(C_i) = S_i = \{s | s \in S, C(s) \subset C_i\}$ that have at least one label associated with C_i. These samples then represent the input to the splatters of the ith subset of the selected root concept C_r. If samples have multiple labels, each sample can serve as input to multiple splatters.

Concept splatters are regions in the 2D latent space projection containing samples of the same concept. The shape of the concept splatters are based on the density of associated samples in the latent space. To define the concept splatters for a subconcept C_i, we use the principle of splatterplots [36] and compute a kernel density estimation for all samples associated with this subconcept S_i:

$$\hat{f}_h(\mathbf{x}, S_i) = \frac{1}{|S_i| \cdot h} \sum_{j=1}^{|S_i|} K\left(\frac{\mathbf{x} - \mathbf{x}(s_j)}{h}\right), \tag{1}$$

where $\mathbf{x}(s_j)$ is the 2D coordinate of the jth sample of S_i after dimensionality reduction, K is a Gaussian kernel, and $h \geq 1$ is an adjustable bandwidth, which controls the smoothness of the estimated probability density function. The shape of the final splatters is defined by an isocontour from the density field using an adjustable density threshold t. The lower the bandwidth h, the lower the level of abstraction, i.e., the more intra-class variability can be observed through an increasing number of separated splatters. Through the density threshold t, the user controls the size and number of visible concept splatters. The higher t, the smaller the splatters, and the more likely that small isolated concept splatters disappear.

Conversely, we can compute a mapping of arbitrary latent space regions onto the concept space. Users manually create regions in the latent space that represent queries for the concept space. A query selects a subset of samples $S_q = \{s | \mathbf{x}(s)$

$\in CS_q, C(s) \subset C_r\}$, where all sample positions of S_q lie within the manual splatter query region CS_q and their labels are leaf nodes of the user-selected root concept C_r. For each concept $c \in C_r$, we can now compute its relevance to the user query simply as the ratio of its associated samples within the query splatter $S_q(c)$ to all its associated samples $S(c)$:

$$r(c, S_q) = |S_q(c)|/|S(c)|. \qquad (2)$$

4.2. Splatter annotation

Our second hypothesis states that a few data samples are sufficient to characterize the content of a splatter. As representative prototype samples, we select one data instance for up to k most relevant labels in the splatter. We define the relevance of a label (and its associated concepts, respectively) for a splatter as its proportional contribution to the splatter content:

$$r(c, i, j) = |S_{ij}(c)|/|S_{ij}|, \qquad (3)$$

where $S_{ij} = \{s|\mathbf{x}(s) \in CS_{ij}, C(s) \subset C_i\}$ is the set of data instances associated with CS_{ij}, i.e., the jth concept splatter of C_i (Fig. 3D).

We select one prototype sample for each label. If a ranking attribute is available, data instances are sorted within $S_{ij}(c)$ according to this attribute. For word embeddings we use vocabulary frequency to sort the instances so that more popular words are ranked higher. If no attribute is given, the underlying data distribution can be used to select representative samples [48,49].

In addition, each concept splatter gets a title. The title corresponds to the lowest common ancestor concept of the contained data instances. In other words, the title corresponds to the hypernym of the contained data. When visualizing thousands of classes, it is unlikely to have spatial regions that are 100% covered by a single concept. We therefore set the threshold for the lowest common ancestor concept to 95% of instances in the splatter. This means that the hypernym title corresponds to the label of the deepest descendant concept $c \in C_i$ with relevance $r(c, i, j) \geq .95$ (Eq. (3)).

4.3. First validation

For a first validation of our hypotheses, we use a simple neural network consisting of two fully connected layers to classify ten different categories of fashion items represented as 28 × 28 pixel-resolution black-and-white images from Fashion MNIST (FMNIST) [13]. For the latent space, the features of the second-to-last layer were extracted based on 70,000 images, from both, the training and the validation dataset.

To validate H1, we investigated whether the observable visual patterns using concept splatters are indeed indicative of quantifiable network properties. As quantification of the visual patterns, we used the simple *class distance consistency* (DSC) score, which measures the ratio of class instances that are located closer to their own class centroid than to the centroid of another class [50]. Our expectation is that classes which the network has learned to separate well should also yield a high DSC score in a similarity map. Indeed, for the 10 classes of FMNIST, there is a strong positive correlation between the DSC and the classification accuracy (Pearson's $r = 0.85$). As the truthfulness of the 2D presentation decreases due to distortions caused by the dimensionality reduction [28], this correlation is slightly lower in the 2D projection than between accuracy and DSC in high-dimensional space (Pearson's $r = 0.91$).

To illustrate the effectiveness of the visual patterns, we show the concept splatters of the class "shirt", which has the lowest accuracy and is often mis-classified with t-shirts and other upper body clothing classes [51] (another case can be found in Section A.1 of the supplemental document). Fig. 4 shows that *shirt* is separated into two splatters, overlapping with splatters of *t-shirt* and *pullover*, respectively. The overlaid heatmap illustrates that mis-classifications are mostly associated with data instances within regions where splatters are overlapping. This shows that overlapping splatters can be indeed a strong indication whether the network has sufficiently learned the expected categorization. In contrast to a confusion matrix (Fig. 4 right), however, concept splatters not only provide hints whether the potential confusion may stem from low inter-class variability but also reveal high intra-class variability and rare categories, which may explain *why* the network tends to confuse the classes.

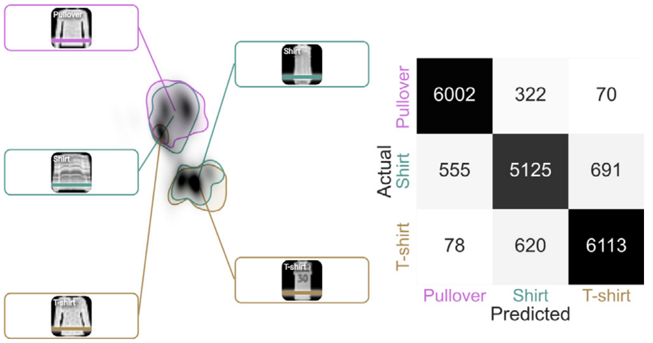
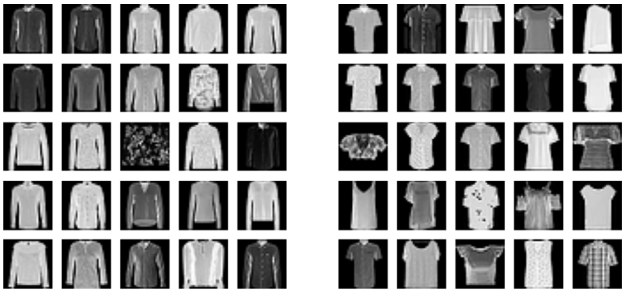


Fig. 4. Heatmap encoding the number of mis-classified data instances on top of concept splatters for *shirt* (green), *t-shirt* (brown) and *pullover* (pink) (left) and a confusion matrix of the same data (right).



(a) Long-sleeved (b) Short-sleeved/sleeve-less

Fig. 5. 25 random samples for each of the two *shirt* splatters in Fig. 4.

Second, we analyzed if separate splatters of the same concept stem from explainable intra-class variability. We again looked at the class label "shirt" and retrieved 25 random images for both *shirt* splatters shown in Fig. 4 (more samples in Section A.2 of the supplemental document). If separate splatters represent an explainable categorization, which is not expressed by the concept space, then the content of each group of images should be able to be characterized by visual attributes that applies to all images within the group, but not to any image of another group. As shown in Fig. 5, the two *shirt* splatters can be clearly characterized by their sleeve length. This illustrates that separated splatters can be interpreted as basic categories a network applies on the data instances — in this case long sleeved vs. short-sleeved/sleeve-less tops. As it is possible to find a characterization that applies to all the random samples of a splatter, but not to any instance of the respective other splatter, it should therefore also be possible to pick any instance of a splatter as prototype sample, as stated in H2.

5. Interaction design

Our goal is to facilitate network diagnosis by an interactive visual investigation of network responses to a large dataset. We

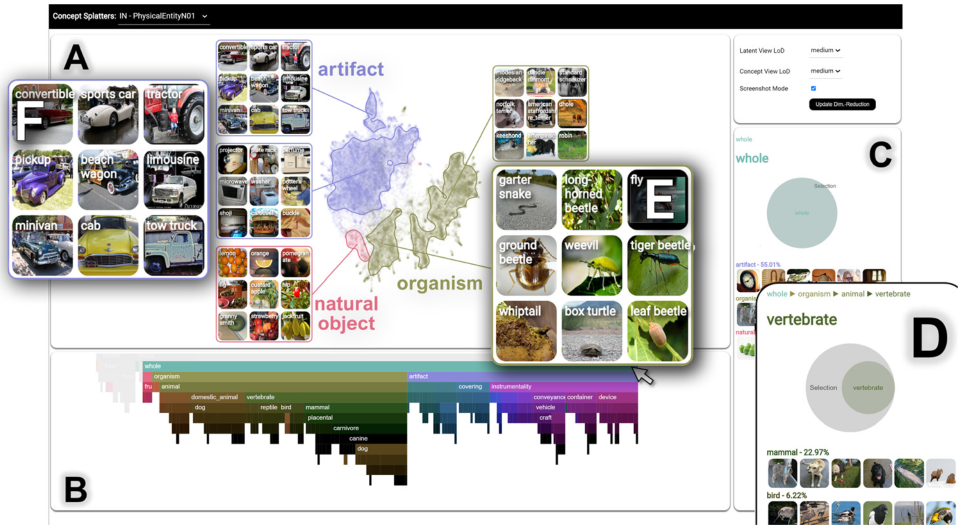


Fig. 6. Concept splatters visualize what the human *and* the machine consider to be similar by showing how a human-understandable concept space (B) maps onto a latent space constructed from activations of 50,000 ImageNet images (A). The detail view shows the current root selection (C) and allows to inspect selected splatters (like the right organism splatter in D) or spatial selections. *We can visually confirm a previous observation by Deng et al. [52] that the network distinguishes between man-made artifacts (purple) and organisms (green), as well as natural objects (pink). In addition, we can observe that the organism concept is separated into invertebrates/reptiles etc. (E) and a second splatter containing vertebrates like mammals and birds (as shown in the detail view in (D)). The enlarged top left inset shows a separated splatter of artifact, which contains only vehicles (F). Please zoom in for more details.*

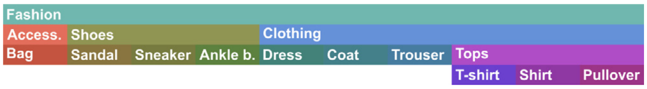


Fig. 7. Concept space used for FMNIST, adapted from the Zalando web page structure.

use the principle of concept splatters explained in Section 4 to support such a diagnosis following the famous information seeking mantra [53] in multiple coordinated views (Fig. 6).

The **concept view** represents the hierarchical, explanatory concept space as an icicle plot. Each bar represents a concept $c \in C$ and its width encodes the number of associated samples $|S(c)|$. This allows users to see the structure of the complete concept space and also to spot important concepts with large numbers of associated data instances. In Fig. 7, we show the hierarchical categorization of fashion items on the Zalando web page as concept space, where each leaf concept has an equal number of associated data instances.

In the **latent view**, each sub-concept of a user-selected concept C_r is represented by one or more concept splatters. We provide two visual representations of concept splatters: They can be shown as bounded and annotated regions on top of a classic color-coded similarity map (as shown in Fig. 3E and Fig. 6A) or as simple curves, similarly to multi-class splatterplots [36] (as shown in Fig. 3C), with annotations. We use simple splatter curves as default view in our web implementation as they are more efficient to render and therefore more suitable for interactive exploration. We allow users to switch to a color-coded similarity map view for creating more visually interesting screenshots.

Visual linking between the concept and latent view is based on a common **color scheme**, where concept splatters are assigned the distinct color of their associated concept C_i. We applied a hierarchical coloring scheme adapted from Tennekes et al. [54]. Our variant assigns larger parts of the hue spectrum to nodes with more child nodes. This has the effect that large nodes with many children get assigned very distinct colors which leads to a better color separation deep down in the hierarchy for unbalanced trees. The resulting color scheme can be seen in Fig. 6B.

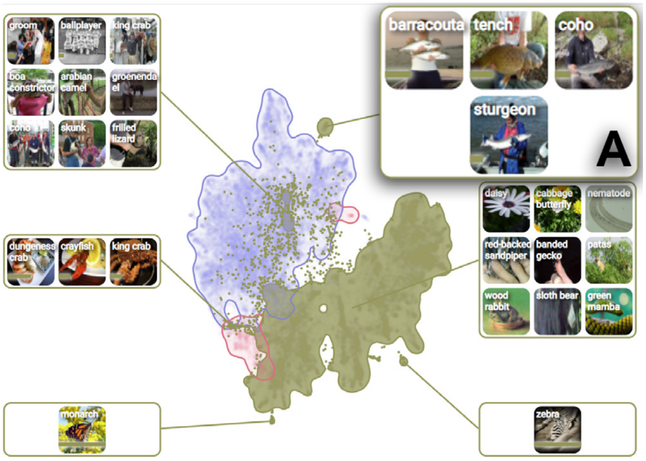


Fig. 8. Selection of sub-concept organism highlights the associated concept splatters and shows their associated insets, as well as scattered outliers. The enlarged inset (A) shows a rare organism category of fish held by humans.

Concept splatter annotations to characterize their content are shown as **insets** attached to the respective splatters. We show a maximum of six insets at a time, where each inset depicts up to $k = 9$ most relevant labels (see Section 4.2). To indicate the prominence of a label for the concept splatter, we encode its relevance (Eq. (3)) as a bar at the bottom of the prototype. The insets are spaced equally at the vertical sides of the latent view, avoiding overlaps with its contents.

5.1. Selections

We support interactive exploration on a fixed level of abstraction through brushing and linking between the concept and latent view. The default selection is C_r, i.e., the root node of the concept space at the chosen level of abstraction.

Users can select a sub-concept of C_r by hovering over the icicle plot cells in the concept view. This **highlights** the corresponding concept splatters in the latent view, and we only show insets for the selected sub-concept (Fig. 8). In the latent view, up to 1000

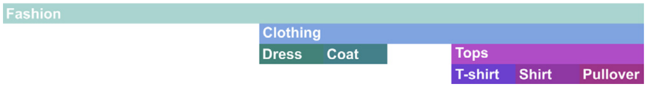
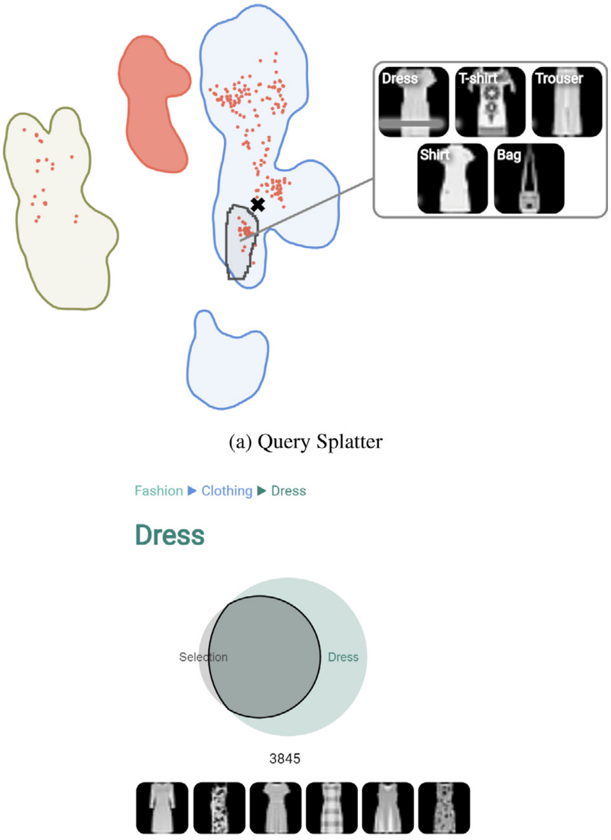


Fig. 9. Heatmap of the concept space in Fig. 7 after hovering the top right *clothing* splatter in Fig. 3. *This splatter contains all types of* clothing, *except for* trousers.



(a) Query Splatter

(b) Detail View

Fig. 10. Query selection around bag outliers in the latent space (a) and the corresponding detail view (b). *The Euler diagram in the detail view shows that dresses make up around 95% of the query's content*.

scattered outlier instances become visible upon a sub-concept selection.

To select a single splatter of a sub-concept, users hover over the desired splatter in the latent view. Then, we only show the inset associated with the selected splatter. The icicle plot visualizes the relevance of each concept for the selected splatter (Eq. (2)). The relevance is encoded through opacity, which creates a **heatmap**-like concept tree representation, as shown in Fig. 9.

Finally, we allow users to select arbitrary query splatters in the latent space through a free-form **lasso-selection**. Employing query splatters, users can inspect the local neighborhood of overlapping sub-concepts to reason about the attributes overlapping concepts have in common. In addition, query splatters allow them to select regions containing outliers that are not covered by concept splatters (see Fig. 10(a)). Like a conventional concept splatter, query splatters are also illustrated by insets showing prototype samples. The query splatter stays active until users manually close it using the x-symbol above it.

To explore the content of selections in more detail, we provide a **detail view**. The detail view initially shows the lowest common ancestor concept of the selection as title (see Section 4.2), as well as up to six samples for each child concept of the user-selected root concept C_r (see Fig. 10(b)). Child concepts with a relevance $\leq 5\%$ according to Eq. (2), are aggregated into a common category *Others*. The users can investigate data instances of the children of each concept by clicking on the concept labels in the detail view. They can also use the breadcrumbs at the top of the view to navigate to a higher hierarchy level. This way, users can use query splatters to select concept outliers that are not captured by the visualized concept splatters and inspect them in the detail view, such as bags and shoes looking like dresses in Fig. 11. We show a Euler diagram to illustrate the relation between the selected root concept in the detail view and the user selection. In Fig. 10, the query splatter consists almost exclusively of dresses, and most dress images lie within this spatial region in the latent space. The few other fashion items in the selection are similarly long-stretched as dresses (Fig. 11).

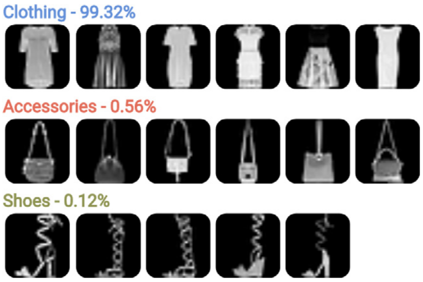


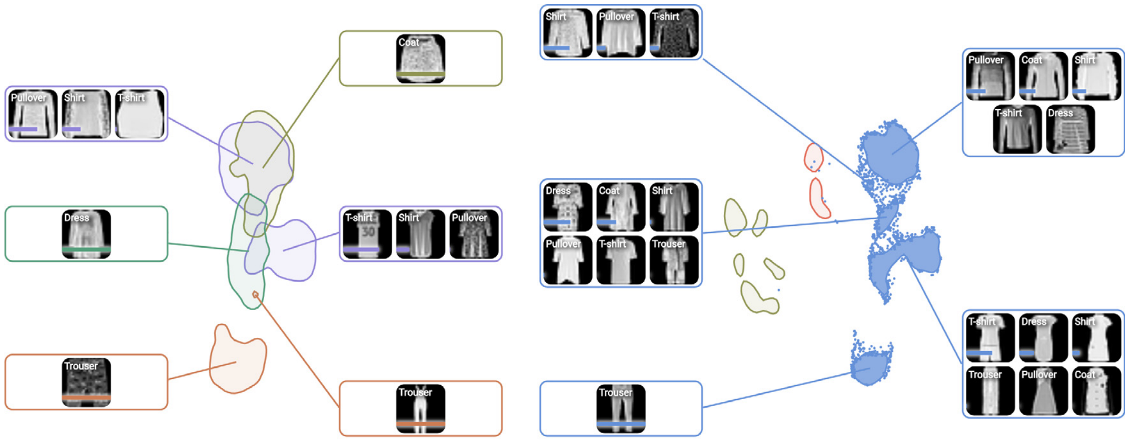
Fig. 11. Samples associated with the parent concept fashion selected from the detail view of Fig. 10(b). *Some bags and shoes look very similar to dresses due to their elongated shapes.*

5.2. Zoom & filter

To effectively inspect very large datasets with a high number of data instances and concepts, we support interactive adjustments of the level of abstraction for both, the concept space and the latent space. Drill-down in concept space can be performed by clicking on a node in the concept view, so that the selected node then represents the new root concept. Both, the concept view and the latent view, are updated, revealing the sub-concept structures. The latent view is filtered so that only data instances associated with the selected sub-concept are shown. Fig. 12(a) shows a drill-down from the overview in Fig. 3 to the *clothing* concept, which is separated into four sub-concepts.

A problem with dimensionality reduction is that many of the neighborhood relationships in the higher-dimensional space cannot be preserved if reducing the data to two dimensions. In lower hierarchy levels of the concept space, this can result into non-existing concept overlaps. For a more truthful inspection and better usage of the display space, we therefore provide the option to **recalculate** the latent space projection only for the samples associated with the selected concept C_r after drill-down (for an example, see Section A.3 in the supplemental document).

To change the level of abstraction in the latent space, we vary the bandwidth and density threshold parameter of the kernel density estimation (Eq. (1)). The lower the bandwidth, the lower the level of abstraction and the more intra-class variability in the latent space can be observed. For example, Fig. 12(b) shows concept splatters of the root *fashion* on a low level of abstraction to observe how the machine sub-categories the dataset. When selecting the sub-concept *clothing*, it can be observed how the machine "sees" most variability based on the overall length of the top and the length of its sleeves.



(a) Drill-Down in Concept Space (b) Drill-Down in Latent Space

Fig. 12. Drill-down approaches from the overview of FMNIST shown in Fig. 3E: selecting *clothing* as root concept (a) or lowering the bandwidth (b).

6. Web-based implementation

We provide concept splatters as web-based tool for the exploratory diagnosis of network behavior. As we let users interactively explore network responses based on a large dataset, the main challenge is to find a balance between interface responsiveness and flexibility by carefully selecting which aspects to precompute. The major bottlenecks are (1) the dimensionality reduction calculation and (2) querying the selected samples after user interaction, such as drill-down, hovering over concepts or concept splatters, and manual splatter queries. Our implementation was tested in Google Chrome on a consumer laptop, with 16 GB of RAM and 4 CPU cores.

Our implementation uses a client–server architecture. The Python server is used to process the data samples and precompute both, concept space and latent space, during the start of the server. Using the networkX library [55], we convert concept space data structures into proper trees by removing potential cycles. For dimensionality reduction of the latent space we use the UMAP implementation by McInnes et al. [44]. We chose 30 neighbors and a minimum distance of 0.2 as hyperparameters to balance preservation of global and local structures (see Section A.4 in the supplemental document for a parameter comparison). Updating the dimensionality reduction is the most time-consuming process and can take several minutes for datasets with hundreds of thousands of samples and hundreds of dimensions [44]. Our server therefore pre-computes UMAP for all data instances at start-up. When users drill into the concept space and thereby filter the instances, they can recompute the dimensionality reduction for a sub-concept on demand, and the server caches the result for later reuse. For the user interface and interaction on the client side, we employ a combination of HTML, CSS, and JavaScript. The visualizations are created with D3.js [56], and the d3-contour library is used to compute the concept splatter geometry.

A key requirement for the interactive exploration of the hierarchical concept space are real-time selections — either of a sub-concept C_i (*concept query*) or a *latent space query* by selecting a single splatter CS_{ij} or making a spatial query S_q using the lasso selection (see Sections 4.1 and 5.1). As the ground truth labels of the data instances are fixed, so is the mapping of the instances to concepts. We therefore optimize concept queries by precomputing the sets of data instances for each concept during the construction of the concept tree. However, latent space queries cannot be precomputed, as the splatter shape and the spatial user selection can be arbitrary. To speed up the selection process, we organize the data instances of the latent space in a 128 × 128 quadtree. For the insets, data instances are ranked to ensure that more popular instances, such as more well-known words, are preferred as prototype samples. This means that data instances need to be sorted across multiple quadtree cells. To make this sorting more efficient, we only save up to nine samples per cell, which is sufficient to generate insets such as shown in Fig. 6. As users can drill down the concept space *and* make spatial queries in the latent space, we have to perform a complex query for S_q, as described in Section 4.1. To speed up this process, we precompute multiple quadtrees for all higher-level concepts in the concept space. In practice, we found that computing quadtrees for all concepts with at least 10% of all data instances provides a good balance between precomputation time, storage, and runtime performance. If users create a query splatter, we consult the quadtree associated with the selected concept C_r or the closest ancestor concept that has a quadtree. As shown in Fig. 13, this approach is beneficial for large latent spaces with more than 10,000 samples, but it leads to a constant overhead of around 10 ms. Tests with an artifically generated dataset on a consumer laptop, as described in Fig. 13, showed that complex queries require around 10 ms for 100,000 samples and slightly below one second for one million samples.

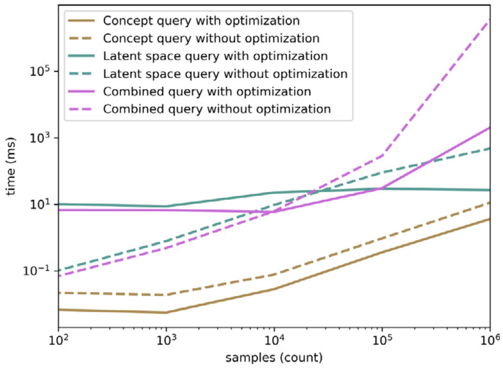


Fig. 13. Comparison of the runtime for concept and latent space selections, as well as their combination for an increasing number of data samples. *Note: The measurements for the not optimized, combined latent and concept query at* 10^6 *samples were terminated after an hour*.

7. Use cases

In the previous sections, our examples were primarily based on FMNIST and a simple neural network (see Section 4.3). To demonstrate the applicability of concept splatters beyond this simple example, our web-based implementation of concept splatters supports three additional scenarios:

ImageNet is an image database containing millions of human-annotated images organized into tens of thousands of WordNet noun synsets. It is the standard benchmark for large-scale object recognition [57]. Latent space representations of ImageNet have been shown as grid of thousands of images based on their dimensionality-reduced activation vectors [23] and as hierarchical scatterplots of landmark samples [9]. Using concept splatters, we analyzed the latent space created by **Inception-V1**, which was trained on the ILSVR 2012 validation dataset [57] with 50,000 images assigned to 1000 labels. As concept space, we use WordNet itself, rooted at *physical entity*. Fig. 6 shows clearly how well neural networks can separate between organisms and man-made objects already on a high level of abstraction. Other interesting observations can be found in Section B of the supplementary document.

It has been shown that pretrained ImageNet weights provide good initial features for many fine-grained image classification tasks [6]. Here, we use concept splatters to visually inspect the transferability of a pretrained network. In our example, we aim to transfer the feature representation learned from ImageNet images to the images of the **Oxford flowers** dataset [58] by freezing the entire convolutional base. This dataset contains 17 classes with 80 images each. In the initial 1000 ImageNet classes, there are only two flower classes, from which only the class *daisy* is included in the Oxford flowers dataset. This means that 16 out of the 17 flower classes have not been seen by the network during training. As concept space, we use the botanical taxonomy, starting from the order of the plants. Fig. 15 shows a high inter- and intra-class variability of the dicotyledons group with a query splatter around overlapping orders. Further drilling down in concept space reveals that individual flower families can be separated quite well (see Section C of the supplementary document).

As a complimentary example to image-based data samples, we also investigated the applicability of concept splatters to explore word embeddings. One of the most wide-spread word embeddings based on **word2vec** was trained on the **Google news** dataset, containing around three million words, each described by a 300-dimensional feature vector [17]. As concept space, we again use WordNet [41] as it captures a very large variety of concepts. We map all words to one or multiple synsets, which leaves us with a subset of around 200,000 words that are known in both spaces. As nouns, adjectives, and verbs do not share a common root node in WordNet, we introduce intermediate nodes for each part of speech and combine them in a common root node. As many words can be associated with multiple synsets, and, conversely, many words can also be associated with the same synset as synonyms, we end up with 103,000 synsets as leaf nodes of the concept space. As prototype samples were often very rare words, we only keep the lemmas of each word. We end up with around 55,000 sample words and 89,000 concept leaf nodes. We use a word2vec vocabulary that is sorted by frequency so that more commonly used words are shown in the insets. Fig. 14 shows how word2vec separates nouns describing abstract concepts and physical entities. Individual splatters thereby contain topically similar words. Further scenarios are shown in Section D of the supplementary document.

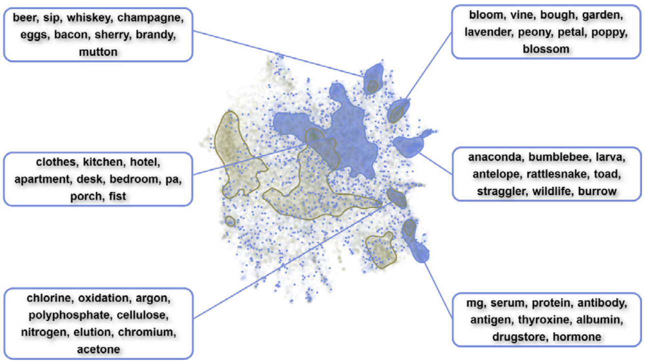


Fig. 14. Exploration of WordNet entities learned by word2vec: Splatters of various physical entities (blue) are revealed besides abstract concepts (green) (in clock-wise order): plants, animals, medical terms, chemical compounds, household items and places, food and drinks. Insets manually enlarged for better readability.

8. Qualitative evaluation

The goal of our work was to provide a scalable solution for similarity maps so that users can visually inspect if the similarities learned by the network correspond to the users' expectations. To evaluate if concept splatters fulfill this goal, we gathered feedback from four users with different professional backgrounds and prior knowledge levels (Table 1). Due to the pandemic situation, three of these feedback sessions were conducted remotely. Depending on the users' professional backgrounds, we let them explore different datasets. All users were first introduced to the concept of similarity maps by demonstrating Tensorflow's embedding projector [18] with the MNIST dataset. Subsequently, we demonstrated the features of concept splatters using our simple FMNIST example. Afterwards, users were encouraged to explore the latent space of their assigned dataset (see Table 1) while thinking aloud. All sessions were screen- and audio-recorded. The audio tracks were transcribed and analyzed through open coding. On average, a session lasted 45 min.

8.1. Observations and discoveries by users

Users utilized the **concept view** to navigate through the concept space and seemed to grasp the hierarchical aggregation of labels immediately. User PS quickly discovered an error in the plant taxonomy, which was fixed before the exploration session continued. Users DS, SC, and MM also soon discovered that WordNet is not a perfect hierarchy because some synsets are associated with more than one parent. In our implementation, these synsets are duplicated, which initially caused slight confusion. A prominent example for a duplicated child node is the synset *dog*, which is descendant of *canine* and *domestic animal* (see Fig. 6B). User MM thereby also discovered a strong imbalance towards dogs in the ImageNet dataset [15], of which he has been previously unaware.

Most discoveries were done in the **latent view**, and most of these discoveries concerned **inter-class variability**. User DS discovered, for instance, that Inception-V1 can clearly separate some types of food, while user MM was impressed about the clear separation between *cat* and *dog*, as well as higher-level animal categories like *fish* and other *vertebrates*. User PS felt confident to conclude that the network had rather successfully learned to separate different flower classes — apart from few exceptions. Most attention was attracted by overlapping splatters indicating low inter-class variability. For instance, SC found similarities between *whale*, *sea lion*, and *megalith*, and DS pointed out striking visual similarities between different musical instruments (e.g., *piano*, *marimba*, and *accordion*). MM specifically explored overlapping splatters between *cat* and *dog* and thereby discovered several visually similar examples or problematic cases, like mislabeled instances. In the flowers dataset, PS spotted an overlap between

Table 1
Users of the qualitative evaluation.

User	Professional background	Prior knowledge	Dataset
DS	PhD student and researcher in data science	Solid ML and Vis background	ImageNet
SC	Professional in science communication	Basic knowledge of ML and Vis	ImageNet
MM	Senior researcher in multimedia retrieval	Deep knowledge of ML, solid knowledge of Vis	ImageNet
PS	Post-doctoral researcher in plant science	No special prior knowledge in ML or Vis	Flowers

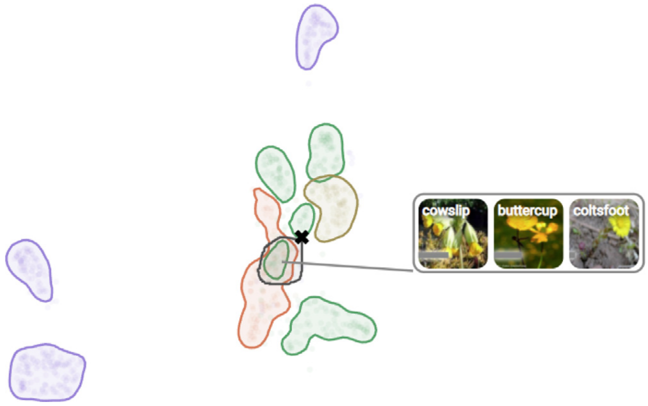


Fig. 15. Overlap between plant orders containing *cowslip* (red) and *buttercup* (green) discovered by user PS.

buttercup and *cowslip* (Fig. 15), as well as a strong similarity between *coltsfoot* and *dandelion*, which is known to be challenging in this dataset [58].

High **intra-class variability** was discussed relatively rarely. One of the few exceptions was user MM who found the separation of organism into two groups, as shown in Fig. 6, notable. **Rare categories** were discussed more often: Among other discoveries, SC spotted a small isolated splatter of fish held by humans (see Fig. 8) – a known spurious learning strategy, where networks have learned to identify the class *tench*, as well as other "trophy fish", based on hands or fingers instead of visual features of the fish itself [5,59].

Users tended to reason about their discoveries based on the examples shown in the insets or the detail view. For instance, PS speculated that the network sometimes focuses too much on colors and camera angles and still too little on the most relevant features, like inflorescence. SC argued that *megalith*, *sea lion*, and *whale* probably share similar texture properties. Similarly, MM explained that some of the *structure* instances overlapping with *organism* share visual resemblance with animal parts, such as instances of *coil* looking like a snail or a *honeycomb* resembling insect eyes. He also explained that some rare categories of *artifact* are ambiguous because they also contain an animal or human. He pointed out that it is known that a lot of ImageNet images could have more than one valid label [60].

More user findings as well as screenshots illustrating these findings can be found in Section E of the supplementary document.

8.2. User feedback

Users intuitively understood and appreciated the notion of hierarchically organized labels. MM noted that, this way, it is possible to explore even very small subclasses. MM and DS were also intrigued by the option to perform spatial selections, especially for exploring overlaps. For that purpose, DS particularly liked the detail view with the display of the lowest common ancestors and the Euler diagram to assess selections. He said that, this way, he could see how splatters overlap and how they are distinct to explore how things are similar and dissimilar. MM also liked the option to drill down in latent space to be able to inspect different types of overlaps. SC explained that, through concept splatters and their insets, one can discover potential confusions he would never think of if he would not see it, such as the strong visual similarity between some fish and airplanes.

Users also mentioned shortcomings and suggestions for improvement. DS and MM pointed out the limited ability of a 2D projection to convey true high-dimensional relations. Both users therefore highly appreciated the ability to recompute the projection based on the current selection. However, SC mentioned that he had slight problems staying oriented as some splatter distances were significantly altered by this recomputation. For instance, the relative distance between trophy fish and the remaining fish was significantly shortened after the recomputation. DS worked a lot with selections and was asking for more elaborate selection methods, such as being able to not only select the common instances between two concepts but also their discriminating aspects. He considered lasso selections sometimes tedious for such queries. Similarly, MM suggested to have richer options to discover and inspect individual outliers apart from lasso selections to spot potential problems in the training data.

As an expert in machine learning, MM pointed out that concept splatters are not only valuable for judging how well networks separate human-interpretable concepts but also to assess the quality of training data. Especially for the second scenario, he appreciated the provided interaction techniques. The key aspect, for him, was how concept splatters solve the "*visual overflow problem*" when analyzing datasets beyond MNIST, which "*is not real world complexity*".

9. Conclusions and future work

In this work, we presented concept splatters as a novel method based on prototype theory [16] to interactively assess networks based on large datasets. We showed that a visual encoding, which maps a pre-defined taxonomy (i.e., a categorization that is expected by the user) onto dense regions within a machine's learned latent space (i.e., the categorization by the network), can reveal if a network has learned the expected categorization. This categorization may capture the ground truth labels of a classification network, such as in our FMNIST and ImageNet scenarios, but any other explanatory, specialized concept space can be used, such as the botanical taxonomy. Through the mapping of the concept space onto the latent space, we can derive a selection of few prototypical data instances, which can provide indications *why* a network categorizes data instances differently than expected. Our web-based implementation can handle image and text data and is sufficiently fast to support smooth interactive exploration of datasets such as ImageNet or vocabulary from the Google news corpus. Using concept splatters, we could generate visualizations illustrating known characteristics of neural networks using large labeled datasets, such as inter-class variabilities that indicate potential misclassifications. Users of our qualitative evaluation could effectively explore which concepts the network could separate well across multiple levels of abstraction using datasets of real-world complexity. Annotated concept splatters enabled them to reason about causes of potential confusion, such as spurious learning strategies, ambiguous class labels, or unexpected visual similarities.

We have presented concept splatters as isolated visualizations and interactive exploration tool. In the future, we see a great potential of concept splatters to serve as effective interaction method to select a group of interesting data instances for further inspection in linked views, for instance through confusion matrices or attribution graphs [5]. Concept splatters can also serve as new underlying mechanism to visually compare network responses to a large dataset across different network architectures, training sets, or along the training progress. While our current approach requires a pre-defined concept space for large numbers of labels, we could investigate data-driven methods to create hierarchical concept spaces for unstructured datasets. Finally, future extensions of concept splatters can go beyond fully labeled datasets by predicting labels of unknown instances — optimally combined with active learning to interactively steer the mapping between latent space and concept space.

Concept splatters combine multiple strategies to address the scalability of similarity maps based on prototype theory — namely hierarchical aggregation in both, latent and concept space, splatter illustration through automatically generated insets, and new selection-based interaction techniques. To better understand how these individual strategies facilitate visual exploration of large latent spaces, it will be necessary to investigate these strategies in controlled user studies in the future.

CRediT authorship contribution statement

Nicolas Grossmann: Software, Data curation, Conceptualization, Methodology, Formal analysis, Writing – original draft. **Eduard Gröller:** Writing – review & editing. **Manuela Waldner:** Conceptualization, Methodology, Validation, Formal analysis, Funding acquisition, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

VRVis is funded by BMVIT, BMWFW, Styria, SFG and Vienna Business Agency in the scope of COMET—Competence Centers for Excellent Technologies (854174) managed by FFG. The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Programme.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cag.2022.04.013.

References

[1] Liu M, Shi J, Li Z, Li C, Zhu J, Liu S. Towards better analysis of deep convolutional neural networks. IEEE Trans Vis Comput Graph 2016;23(1):91–100.

[2] Rauber PE, Fadel SG, Falcao AX, Telea AC. Visualizing the hidden activity of artificial neural networks. IEEE Trans Vis Comput Graph 2016;23(1):101–10.

[3] Kahng M, Andrews PY, Kalro A, Chau DHP. Activis: Visual exploration of industry-scale deep neural network models. IEEE Trans Vis Comput Graph 2017;24(1):88–97.

[4] Pezzotti N, Höllt T, Van Gemert J, Lelieveldt BP, Eisemann E, Vilanova A. Deepeyes: Progressive visual analytics for designing deep neural networks. IEEE Trans Vis Comput Graph 2017;24(1):98–108.

[5] Hohman F, Park H, Robinson C, Chau DHP. Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations. IEEE Trans Vis Comput Graph 2019;26(1):1096–106.

[6] Kornblith S, Shlens J, Le QV. Do better imagenet models transfer better? In: IEEE CVPR. 2019, p. 2661–71.

[7] Xiang S, Ye X, Xia J, Wu J, Chen Y, Liu S. Interactive correction of mislabeled training data. In: IEEE VAST. 2019, p. 57–68.

[8] Oh Song H, Xiang Y, Jegelka S, Savarese S. Deep metric learning via lifted structured feature embedding. In: IEEE CVPR. 2016, p. 4004–12.

[9] Pezzotti N, Höllt T, Lelieveldt B, Eisemann E, Vilanova A. Hierarchical stochastic neighbor embedding. In: Computer graphics forum. Vol. 35. (3):Wiley Online Library; 2016, p. 21–30.

[10] Sainburg T, Thielk M, Gentner TQ. Latent space visualization, characterization, and generation of diverse vocal communication signals. Cold Spring Harbor Laboratory; 2019, BioRxiv. 870311.

[11] Sangkloy P, Burnell N, Ham C, Hays J. The sketchy database: learning to retrieve badly drawn bunnies. ACM Trans Graph 2016;35(4):1–12.

[12] Zheng Z, Zheng L, Yang Y. A discriminatively learned CNN embedding for person reidentification. ACM Trans Multimedia Comput Commun Appl 2017;14(1):1–20.

[13] Xiao H, Rasul K, Vollgraf R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. 2017, arXiv:cs.LG/1708.07747.

[14] Ware C. Information visualization: perception for design. Morgan Kaufmann; 2019.

[15] Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. In: IEEE CVPR. Ieee; 2009, p. 248–55.

[16] Rosch E, Mervis CB, Gray WD, Johnson DM, Boyes-Braem P. Basic objects in natural categories. Cogn Psychol 1976;8(3):382–439.

[17] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. 2013, p. 3111–9.

[18] Smilkov D, Thorat N, Nicholson C, Reif E, Viégas FB, Wattenberg M. Embedding projector: Interactive visualization and interpretation of embeddings. In: NIPS 2016 workshop on interpretable machine learning in complex systems. 2016.

[19] Heimerl F, Gleicher M. Interactive analysis of word vector embeddings. In: Computer graphics forum. Vol. 37. (3):Wiley Online Library; 2018 p. 253–65.

[20] Liu Y, Jun E, Li Q, Heer J. Latent space cartography: Visual analysis of vector space embeddings. In: Computer graphics forum. Vol. 38. (3):Wiley Online Library; 2019, p. 67–78.

[21] Dix A, Ellis G. By chance enhancing interaction with large data sets through statistical sampling. In: Proceedings of the working conference on advanced visual interfaces. 2002, p. 167–76.

[22] Ming Y, Xu P, Cheng F, Qu H, Ren L. Protosteer: Steering deep sequence model with prototypes. IEEE Trans Vis Comput Graph 2019;26(1):238–48.

[23] Karpathy A. t-SNE visualization of CNN codes. 2014, Online https://cs.stanford.edu/people/karpathy/cnnembed/. [Accessed February-2022].

[24] Carter S, Armstrong Z, Schubert L, Johnson I, Olah C. Activation atlas. Distill 2019;4(3):e15.

[25] Chen C, Yuan J, Lu Y, Liu Y, Su H, Yuan S, et al. Oodanalyzer: Interactive analysis of out-of-distribution samples. IEEE Trans Vis Comput Graph 2020;27(7):3335–49.

[26] Wenskovitch J, Crandell I, Ramakrishnan N, House L, North C. Towards a systematic combination of dimension reduction and clustering in visual analytics. IEEE Trans Vis Comput Graph 2017;24(1):131–41.

[27] Poco J, Etemadpour R, Paulovich FV, Long T, Rosenthal P, Oliveira Md, et al. A framework for exploring multidimensional data with 3d projections. In: Computer graphics forum. Vol. 30. (3):2011, p. 1111–20.

[28] Nonato LG, Aupetit M. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. IEEE Trans Vis Comput Graph 2018;25(8):2650–73.

[29] Nguyen A, Yosinski J, Clune J. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. In: Visualization for deep learning workshop at ICML 2016. 2016.

[30] Paulovich FV, Minghim R. Hipp: A novel hierarchical point placement strategy and its application to the exploration of document collections. IEEE Trans Vis Comput Graph 2008;14(6):1229–36.

[31] Rathore A, Chalapathi N, Palande S, Wang B. TopoAct: Visually exploring the shape of activations in deep learning. Comput Graph Forum 2021;40(1):382–97.

[32] Ventocilla E, Martins RM, Paulovich F, Riveiro M. Scaling the growing neural gas for visual cluster analysis. Big Data Res 2021;26:100254.

[33] Fritzke B, et al. A growing neural gas network learns topologies. Adv Neural Inf Process Syst 1995;7:625–32.

[34] Carr DB, Littlefield RJ, Nicholson W, Littlefield J. Scatterplot matrix techniques for large N. J Amer Statist Assoc 1987;82(398):424–36.

[35] Lampe OD, Hauser H. Interactive visualization of streaming data with kernel density estimation. In: 2011 IEEE pacific visualization symposium. 2011, p. 171–8.

[36] Mayorga A, Gleicher M. Splatterplots: Overcoming overdraw in scatter plots. IEEE Trans Vis Comput Graph 2013;19(9):1526–38.

[37] Pezzotti N, Lelieveldt BP, van der Maaten L, Höllt T, Eisemann E, Vilanova A. Approximated and user steerable tSNE for progressive visual analytics. IEEE Trans Vis Comput Graph 2016;23(7):1739–52.

[38] Elmqvist N, Fekete J-D. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. IEEE Trans Vis Comput Graph 2009;16(3):439–54.

[39] Höllt T, Vilanova A, Pezzotti N, Lelieveldt BP, Hauser H. Focus+ context exploration of hierarchical embeddings. In: Computer graphics forum. Vol. 38. (3):2019, p. 569–79.

[40] El-Assady M, Kehlbeck R, Collins C, Keim D, Deussen O. Semantic concept spaces: Guided topic model refinement using word-embedding projections. IEEE Trans Vis Comput Graph 2019;26(1):1001–11.

[41] Miller GA. WordNet: an electronic lexical database. MIT Press; 1998.

[42] Fu R, Guo J, Qin B, Che W, Wang H, Liu T. Learning semantic hierarchies via word embeddings. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: long papers). 2014 p. 1199–209.

[43] Maaten Lvd, Hinton G. Visualizing data using t-SNE. J Mach Learn Res 2008;9(Nov):2579–605.

[44] McInnes L, Healy J, Melville J. UMAP: Uniform manifold approximation and projection for dimension reduction. 2018, arXiv:1802.03426.

[45] Lapuschkin S, Wäldchen S, Binder A, Montavon G, Samek W, Müller K-R. Unmasking clever hans predictors and assessing what machines really learn. Nature Commun 2019;10(1):1–8.

[46] Becht E, McInnes L, Healy J, Dutertre C-A, Kwok IW, Ng LG, et al. Dimensionality reduction for visualizing single-cell data using UMAP. Nature Biotechnol 2019;37(1):38.

[47] Espadoto M, Martins RM, Kerren A, Hirata NS, Telea AC. Toward a quantitative survey of dimension reduction techniques. IEEE Trans Vis Comput Graph 2019;27(3):2153–73.

[48] Joia P, Petronetto F, Nonato LG. Uncovering representative groups in multidimensional projections. In: Computer graphics forum. Vol. 34. (3):Wiley Online Library; 2015, p. 281–90.

[49] Kim B, Khanna R, Koyejo OO. Examples are not enough, learn to criticize! criticism for interpretability. Adv Neural Inf Process Syst 2016;29.

[50] Sips M, Neubert B, Lewis JP, Hanrahan P. Selecting good views of high-dimensional data using class consistency. In: Computer graphics forum. Vol. 28. (3):Wiley Online Library; 2009, p. 831–8.

[51] Seo Y, Shin K-s. Hierarchical convolutional neural networks for fashion image classification. Expert Syst Appl 2019;116:328–39.

[52] Deng J, Berg AC, Li K, Fei-Fei L. What does classifying more than 10,000 image categories tell us? In: European conference on computer vision. Springer; 2010, p. 71–84.

[53] Shneiderman B. The eyes have it: A task by data type taxonomy for information visualizations. In: Proceedings 1996 IEEE symposium on visual languages. IEEE; 1996, p. 336–43.

[54] Tennekes M, de Jonge E. Tree colors: color schemes for tree-structured data. IEEE Trans Vis Comput Graph 2014;20(12):2072–81.

[55] Hagberg A. NetworkX. 2005, Online https://networkx.github.io. [Accessed February-2022].

[56] Bostok M. D3.js. 2011, Online https://d3js.org. [Accessed February-2022].

[57] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. Imagenet large scale visual recognition challenge. Int J Comput Vis 2015;115(3):211–52.

[58] Nilsback M-E, Zisserman A. A visual vocabulary for flower classification. In: IEEE CVPR. Vol. 2. IEEE; 2006, p. 1447–54.

[59] Brendel W, Bethge M. Approximating CNNs with bag-of-local-features models works surprisingly well on ImageNet. In: International conference on learning representations. 2018.

[60] Tsipras D, Santurkar S, Engstrom L, Ilyas A, Madry A. From imagenet to image classification: Contextualizing progress on benchmarks. In: International conference on machine learning. PMLR; 2020, p. 9625–35.