Exploring Visual Cues for Design Analogy: A Deep Learning Approach

Zijian Zhang
Department of Aerospace and Mechanical Engineering,
University of Southern California,
Los Angeles, CA 90089
e-mail: zijianz@usc.edu

Yan Jin[1]
Department of Aerospace and Mechanical Engineering,
University of Southern California,
Los Angeles, CA 90089
e-mail: yjin@usc.edu

The goal of this research is to develop a computer-aided visual analogy support (CAVAS) framework to augment designers' visual analogical thinking by stimulating them by providing relevant visual cues from a variety of categories. Two steps are taken to reach this goal: developing a flexible computational framework to explore various visual cues, i.e., shapes or sketches, based on the relevant datasets and conducting human-based behavioral studies to validate such visual cue exploration tools. This article presents the results and insights obtained from the first step by addressing two research questions: How can the computational framework CAVAS be developed to provide designers in sketching with certain visual cues for stimulating their visual thinking process? How can a computation tool learn a latent space, which can capture the shape patterns of sketches? A visual cue exploration framework and a deep clustering model CAVAS-DL are proposed to learn a latent space of sketches that reveal shape patterns for multiple sketch categories and simultaneously cluster the sketches to preserve and provide category information as part of visual cues. The distance- and overlap-based similarities are introduced and analyzed to identify long- and short-distance analogies. Performance evaluations of our proposed methods are carried out with different configurations, and the visual presentations of the potential analogical cues are explored. The results have demonstrated the applicability of the CAVAS-DL model as the basis for the human-based validation studies in the next step. [DOI: 10.1115/1.4055623]

Keywords: computational support for visual analogy making, visual similarity, unsupervised deep learning, design by analogy, sketching, fixation, artificial intelligence, data-driven design, design representation, design visualization, machine learning

1 Introduction

In engineering design, mental stimulation is useful to boost innovative solutions for ill-defined design problems. During conceptual design, designers, especially novices, usually struggle to choose among various sources to gain insights when attempting to generate creative concepts. In our previous work, it has been shown that the shapes and structures, in addition to behaviors, of a design artifact tend to be more stimulating than the functions [1]. Researchers have observed that designers often search intensively for images from various websites for inspiration [2,3]. Most existing design-dedicated analogy search tools and methods [4–6] require designers to initiate a search by entering *keywords* and using semantic-based approaches for fixation avoidance. Few computational tools exist to support design-by-analogy based on the *visual* similarity analysis. The core research problem in this article is to explore *the roles of computational support* for visual analogy and investigate *how to learn visual features from raw image data and discover potential short- and long-distance analogies* based on visual similarities.

The overall goal of this research is to develop computer-aided visual analogy support (CAVAS) framework that can augment designers' visual analogical thinking by stimulating them by providing various relevant visual cues. Two steps are needed to reach this goal. The first is to develop a flexible computational framework that can explore various visual cues, i.e., shapes or sketches, based on relevant datasets, and the second is to conduct human subject-based behavioral studies to validate the effectiveness and efficiency of such visual cue exploration tools. This article reports the first step of this research together with the overall CAVAS framework description.

Sketching is an efficient way for designers to have their brief and ambiguous ideas taking shapes on paper [7]. The briefness accelerates the transformation of a rough thought into a reality. The ambiguity of an open-ended visual representation contributes to more possible interpretations. Sketching in conceptual design primarily provides potentially meaningful clues for a designer to infer emerging design concepts [8,9]. The inspiration for sketches mostly comes from the *shapes* and the relationships among them. Designers can manipulate given shapes in imagery and combine them into meaningful and even new concepts in a short time. Sketching can reflect premature design ideas in designers' minds, and it is also an ideal stimulant to facilitate creative idea generation. Therefore, the first question for this research is: *How can the computational framework CAVAS be developed to provide designers in sketching with certain visual cues for stimulating their visual thinking process?*

Research has been done to investigate visual analogies in the field of design. Goldschmidt and colleagues demonstrated that visual analogy is considered an effective cognitive strategy to stimulate designers to create innovative concepts for solving ill-structured design problems [10–12]. For novel idea generation, the use of visual stimuli outperforms words [13,14]. In design, shapes may represent semantic concepts and objects to reflect designers' understanding of the visual world. In this article, we assume one sketch only, which includes one shape or one object. A sketch category is the name or the label of the shape or object. From a cognitive point of view, when making a visual analogy, designers can map shapes from high (geometric) dimensions to low (symbolic, conceptual) dimensions [15,16]. At low dimensions, they are capable of interpreting and detecting the similarities between shapes in the

[1]Corresponding author.

Contributed by the Design Theory and Methodology Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received March 1, 2022; final manuscript received August 31, 2022; published online October 6, 2022. Assoc. Editor: Mark Fuge.

same or different categories. It means that designers can abstract perceptual information to some shape patterns, which represent the shape features in a cognitive space [17]. In that space, they can manipulate and transform shapes by exploiting their domain knowledge. From an engineering design point of view, the high-dimensional geometric features signify the lower-dimensional semantic features [18,19], meaning that the high-dimensional shape features can be reduced to a space of a low dimensionality that still preserves the underlying patterns, constraints, and configurations. It is more efficient to explore and exploit the low-dimensional design space to discover novel designs. In the same spirit, computationally transforming *high-dimensional* image sketches (e.g., 2304) represented as *pixels* into *low-dimensional* ones (e.g., 128) captured as *features* can, on the one hand, keep the underline shape patterns of the sketches and, on the other hand, allow the efficient computational shape analysis. In this article, we call the "low-dimensional" space a *latent space*. Therefore, the second question for this research is: *How can a computation tool learn a latent space that can capture the shape patterns of sketches from multiple categories?*

The precondition for making a visual analogy is a visual similarity existing between the source and target domains [2]. In most research on searching for visual stimuli, the magnitude of visual similarity is qualitatively determined by designers [20–22]. The notion of distance is central to measuring visual similarity. Short-distance analogies occur when the source concept is very similar to the target concept; long-distance analogies occur when the source concept is very different from the target concept. The distance can be measured in a feature (i.e., latent) space, representing the dataset. In the latent space, sketches are distributed based on their shape features. Clustering is an essential data analysis and visualization tool and provides a way to *group* sketches in the latent space based on visual similarity. The traditional way of using a deep neural network for clustering images is to train the model for extracting shape features first and then apply clustering algorithms on the extracted features into group images. This leads to a possible mismatch between the learned shape features in the latent space and the clustering of shape patterns and hence the inferior quality of visual analysis [23]. Therefore, the third research problem in this article is that given a latent space for representing shape features from raw pixels, *how can a tool properly cluster image sketches into different shape groups based on their inherent shape patterns and analyze the short- and long-distance analogies based on shape similarity?*

In this article, we apply unsupervised deep learning techniques to build a model, called *CAVAS through deep learning*, or *CAVAS-DL* for short, to learn a low-dimensional latent space, in which shape patterns can be found to distill shape features of the sketches from multiple categories. A clustering layer is constructed to directly cluster images in the latent space during, instead of after, the training process. The distance- and overlap-based similarities are applied to quantitatively measure visual relationships between one category and other categories in the latent space. Short- and long-distance analogies for each category are determined based on the visual similarity metrics. Besides, the connections between different groups of categories are identified to explore how visual analogies can happen.

2 Related Work

2.1 Computational Tools for Design by Analogy. Design-by-analogy consists of two main steps: retrieving potentially inspirational information in the source domains and mapping the inspirational information from source domains to the target domain [24]. Designers often face difficulties when retrieving fitting inspirational sources. Therefore, using effective searching and retrieving tools have the potential to enhance design-by-analogy. A large number of inspirational resources available in various databases can benefit designers who have limited domain knowledge. Many computational tools and methods have been developed to support and enhance searching and retrieval in design-by-analogy. The goals are to strengthen designers' abilities and reduce the influence of experience gaps. Currently, biological systems and patents are two major inspiration sources for design-by-analogy.

Biological systems provide a fruitful source of inspiration for engineering design. Vincent and Mann proposed bio-TRIZ, which adds biological information and principles to the TRIZ database [25]. Chakrabarti et al. created an automated analogical tool called IDEA-INSPIRE that searches relevant ideas from a biological database to solve a given design problem [26,27]. Shu et al. used the natural language analysis to correlate functional basis terms with useful biological keywords [28,29]. DANE (Design by Analogy to Nature Engine) was proposed by Goel et al. to search and retrieve the functioning of biological systems in the structure-behavior-function library [30,31]. Nagel and Stone put forward a computational method to generate biologically inspired concepts based on function-based design tools [32]. ASKNATURE is a web-based tool to interactively classify biological information in the Biomimicry Taxonomy [33].

Patent databases can offer enormous cross-domain technical knowledge to inspire designers. Various computational tools and methods have been proposed to retrieve and analyze patents to support design-by-analogy. Murphy et al. proposed a search methodology to identify inspiring patents, which have functional similarities with design problems [34]. A computation method was put forward for clustering patents based on their functional and surface similarity; then, designers can automatically retrieve analogical stimuli from these patents [35]. As many patent retrieval computational tools focus on mining patents generally, Song and Luo proposed a data-driven method to retrieve patents precisely related to a specific product [36]. Fu et al. proposed a technological distance to measure the "near" and "far" analogical stimuli based on the relative similarity of clusters of patents [37].

While the research into searching and retrieving analogies from biological systems and patents is prolific, the foundation of most research is in linguistics and semantic transfer for analogical reasoning. There are few computational methods and tools that support and guide visual analogy. Luo and coworkers put forward visual analogy support tools based on visual maps of technology domains or technical concepts to guide the search for inspirations across domains or assist the analogical inference from concepts to concepts [38–41]. However, the big difference between the visual cues in this article with theirs is that our visuals are the images and graphics, whereas their visuals are the structures of relations among semantic constructs and design domains.

2.2 Visual Analogy in Engineering Design. CAD, sketches, photographs, and line drawings are the major visual sources that promote analogical thinking [2,42]. In engineering design, many researchers used a large assortment of visual displays to stimulate designers to generate creative design concepts. Jin and Benami indicated that meaningfulness and relevance are the two overwhelmingly important creative properties of visual stimuli that influence design stimulation [1]. Yang et al. showed that the quality and realism of the design can be improved when sketching during concept generation [8,43]. Goldschmidt et al. demonstrated that visual stimuli are useful for both expert and novice designers to improve the quality of design and are more effective for novice designers [11,12]. Linsey et al. illustrated that designers often prefer visual representations to textual descriptions for idea generation, and photographs are growing in popularity due to easy retrieval from the Internet [44,45]. McKoy et al. showed that novice designers can generate higher quality and more novel design concepts when presented with sketches rather than text-based examples [46].

However, displays of visual representations are less effective in producing creative design than reasoning by visual analogy.

Casakin et al. found that if no instructions or directions are provided to guide visual analogy, the quality of the design solutions is mostly diminished [47,48]. It is often said that designers think more visually in their working environment. Designers are more likely to take advantage of shapes and forms of visual displays as stimuli to tackle given design problems [10]. Shape emergence means unexpected or implicit shape features and relations appear only after the manipulation and transformation of explicit shapes [15]. Visual imagery may provide a theoretical foundation for shape emergence in design by linking shape perceptions and cognitive processes of visual reasoning. Therefore, designers often take advantage of visual imagery to reinterpret and reformate underlying shapes from the visual stimuli for idea generation. The precondition for shape emergence is shape ambiguity, which refers to the existence of numerous interpretations of the visual representation [49].

Designers are prone to use sketches to represent rough ideas and obtain hints from the shapes of sketches [7]. The sketch is an informal visual representation that has the property of ambiguity. Because of this property, designers can perceive two or more different shapes from one single sketch. The power of visual analogy is that the designers making the analogy can see the similarities of underlying shapes despite the differences in superficial shapes. Therefore, sketches are an ideal source to serve as a visual stimulus. Recently, the authors introduced computational tools to apply visual analogy in engineering design [50,51]. How to effectively support visual analogy from sketches remains to be a major research question in the design research community.

2.3 Deep Learning Models for Sketch Representation. Recent advances in deep neural network models drastically increased computers' ability to learn a common and general feature space for sketches and images [52–54]. Karimi et al. used a supervised learning method to learn the feature vectors of sketches given the category labels and then create clusters of visually similar sketches based on the learned feature vectors [55]. Jiang et al. introduced a supervised convolutional neural network (CNN)-based approach for patent image vectorization to support visual design stimuli retrieval in design-by-analogy [56,57]. However, in our research, the goal is to learn a latent space that represents the object shape features by using only lines and curves in the sketches rather than having the labels of categories. Therefore, an unsupervised learning approach is needed. *Sketch-rnn* is an unsupervised learning model based on Variational AutoEncoder (VAE) for constructing stroke-based drawings of common objects; it can mimic how humans sketch and draw similar but unique objects [58,59]. *Sketch-rnn* uses a bidirectional recurrent neural network (RNN) [60] as an encoder to capture the features of training data in a latent space Z (e.g., the feature distribution of training data) and applies an autoregressive RNN [61] as a decoder to reconstruct data via a sampled vector z from Z. It means all training data can be mapped to a latent space Z, which can capture shape features. However, the performance of *sketch-rnn* to extract shape features of objects from multiple categories is not satisfactory [23,58,62]. Therefore, a new *sketch-rnn* is needed to robustly present underlying shape features of multicategory objects in a latent space, which can support the measurement of shape similarity.

In summary, a rich body of research on *design by analogy* has yet to be expanded by integrating the extensive work on *visual analogy* and advanced *deep learning* technologies. Our goal in this article is to fill the gap in the three research areas by developing a computational method that can learn the visual similarity from sketches and provide highly effective visual stimuli to enhance the visual analogy of designers.

3 CAVAS: A Visual Analogy Support Framework

Creative designers usually employ inspirational sources that are not directly linked to the design problem at hand, take advantage of incidentally presented cues, and tend to collect a wide range of ideas, sometimes seemingly irrelevant and highly dissimilar, which may lead to insights. Divergent thinking helps designers imagine the world from multiple perspectives, see problems in new ways, and escape stereotypical thinking. There is significant anecdotal and experimental evidence [2,12,63] for the importance of visual analogy to stimulate the originality and creativity of designers. Simply trying to think of or reason analogies and analogous domains is difficult even for experienced engineers. One of the main principles for enhancing analogical reasoning is to provide a variety of related effective cues. Imagine, for instance, a designer is undertaking a concept car design project and wants to employ other domains' styles or technology but is unsure of which to use. In this case, the designer will need to retrieve several short- and long-distance analogy domains to a car, based on the visual similarity from his/her mind and from anywhere possible when his/her mind is not enough.

3.1 Major Functions. Following our previous work on the *generate-stimulate-produce* (GSP) model of creative stimulation [1], a process of called CAVAS can be introduced, as shown in Fig. 1. A designer initiates his/her design process by starting sketching. When the designer carries out the design alone, as shown in Fig. 1(*a*), the sketches the designer generated will be perceived by the designer, hence visually stimulating the designer and leading to further cognitive processes, such as *association* or *analogy*. The results of the cognitive processes will be the production of more design operations, such as *sketching*, which will then generate more *sketches* as design entities. The GSP process keeps going on as design ideas become clearer and design concepts are solidified.

The computer support in the proposed CAVAS is based on a human–computer interaction framework, in which the role of the computer is defined as "to provide highly *relevant* and *stimulating* visual cues to the designer at the right timing during the early idea shaping stage of design." As shown in Fig. 1(*b*), for a computer system, called the CAVAS system, to fulfill this role, it must possess the following six major functions, namely, *learn, analyze, generate, extract, search, retrieve*, and *present*.

***Learn* and *analyze* previous designs from all available sources:** The previous design materials such as sketches, CAD drawings, photographs, and line drawings in the open-source datasets are collected and converted into images. The visual patterns of these images can be learned and represented by the CAVAS system. Then the system can analyze the visual similarity between different domains based on the learned representations.

***Generate* visual analogy databases:** After learning and analyzing previous designs, the CAVAS system can generate visual knowledge in visual and textual formats, which captures the shape patterns of and similarity relationships among the visual components. The generated knowledge is stored in one or multiple visual analogy databases, which can be reused and updated.

***Extract* essential shape information, *search* and *retrieve* visual analogies:** The sketches drawn by designers are fed into the CAVAS system. The system can extract and represent the essential shape information from the sketches and then search and retrieve the relevant visual cues stored in the visual analogy database.

***Present* relevant visual cues:** After the relevant visual cues are retrieved from the visual analogy database, the CAVAS system then presents the visual cues to designers in the ways that the designers are stimulated to find appropriate source analogies from their memory and external databases. The visual cues should increase the chances for designers to retrieve relevant visual analogies. The system can present to designers several short- and long-distance analogy domains to the target design domain based on visual similarities.

3.2 Visual Augmentation Processes. Among the major functions in the CAVAS framework described earlier, *learn* and *analyze* functions are the key ones. Figure 2 shows the entire visual

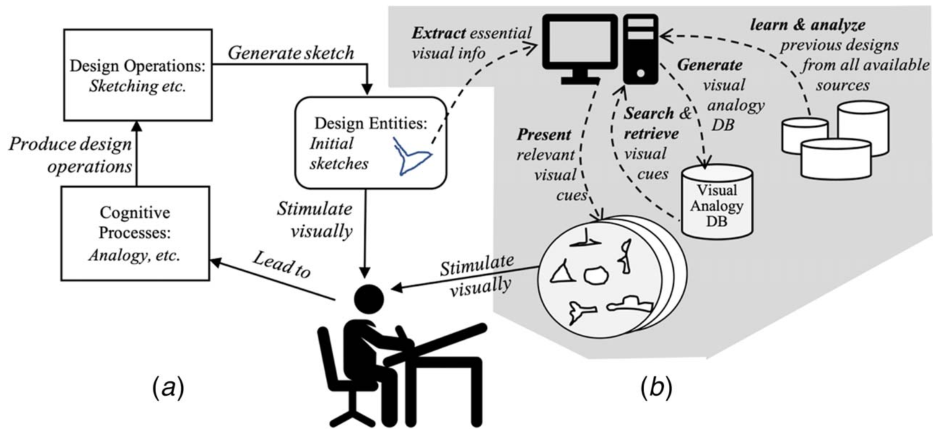


Fig. 1 An illustration of the proposed computer-aided visual analogy support (CAVAS) in a human–computer interaction framework: (*a*) designer's thinking process and (*b*) computer stimuli generation process

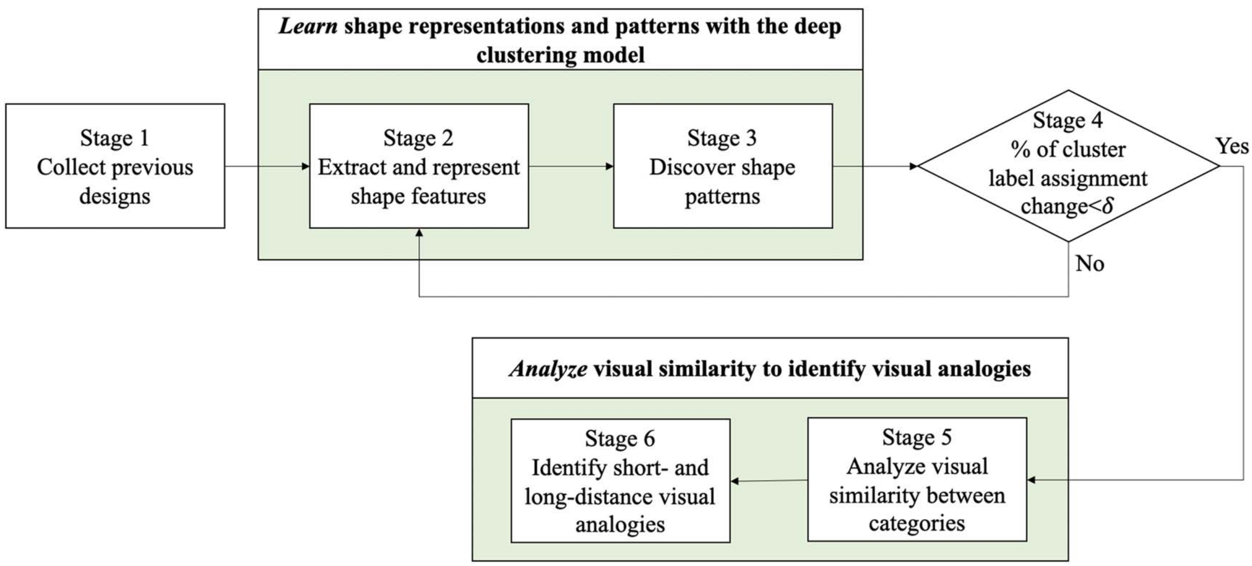


Fig. 2 An entire process of learn and analyze functions in the CAVAS framework

augmentation process, which consists of two main functions and six stages. Each stage is explained as follows.

In *stage 1*, sketches are collected as the previous designs. In this research, the visual cues to be used as visual stimuli are identified based on shape similarities. Sketches made by people often offer various opportunities for interpretation and/or self-reflection. In the eyes of a particular viewer, a sketch could bear a resemblance to an object, person, animal, texture, or place. This ability of cross-domain transformation of shapes can provide a degree of diversity, ambiguity, and uncertainty in the information gathering and idea generation process, which makes it possible for designers to seek inspirations different from their original domain area, e.g., a car designer considers trends in the design of boats. Therefore, sketches are the ideal sources to discover visual cues to enhance designers' visual analogy.

One challenge in augmenting human visual analogy is to make the computer "understand" the sketches (or images) and supply the relevant visual cues to the designer when needed. Since computer images are represented as pixels, given a sketch of 48×48 pixels, a black-and-white image can take a space of 2304 dimensions. For grayscale and color sketches, the dimension size can easily rise to as high as *tens or hundreds of thousands*. In *stage 2*, a dimension reduction approach is taken. Instead of identifying similar sketches in the enormously high-dimensional pixel space, a relatively small number of *shape features* are identified that form a smaller dimensional space for representing the sketches collected in stage 1. Once this shape feature-based space, called *latent space*, is established, it becomes computationally feasible to analyze the sketches to provide relevant visual cues to the designers. It is worth mentioning that the best sets of shape features can be identified by learning from the given datasets collected in stage 1.

The inherent *shape patterns* of collected sketches can be discovered by analyzing and comparing their *shape features* in the latent space. In *stage 3*, a soft clustering approach is taken to cluster the sketches into different shape clusters or groups (the nouns *cluster* and *group* are used interchangeably in this article to indicate the result of the clustering process), i.e., shape patterns, based on their relative "distances" in the latent space. Instead of the "yes or no" designation of a sketch to a given group, each sketch is assigned different probabilities of belonging to multiple groups. This softness preserves ambiguity, which is essential for supporting designers' visual analogy [1]. It is assumed that (1) visually similar shapes should be clustered in the same group to represent one shape pattern, and (2) the sketches of different categories that belong to the same group can be more effective in stimulating designers' analogical thinking due to the shape similarity. These assumptions are established for us to develop the visual cue exploration methodology, and subsequent human-based behavioral studies will be carried out to check their validity.

As the clustering process converges, the size of each cluster becomes stable. In *stage 4*, a ratio is calculated based on dividing

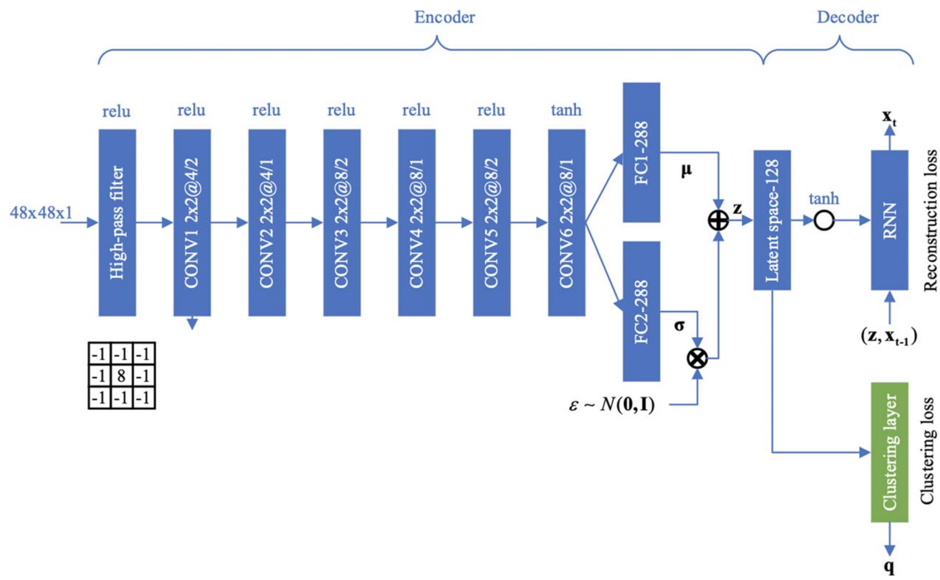


Fig. 3 Structure of CAVAS-DL

the number of cluster assignment changes by the total number of sketches. If it is smaller than the predefined threshold δ, then exit the learning process and jump to stage 5; otherwise, proceed to stage 2.

During the process of providing visual analogy support, the CAVAS system extracts a designer's design sketch information, searches for *relevant* visual cues, and then presents the visual cues to the designer in stimulating ways. The *relevance* here is determined by the similarity measures. In *stage 5*, two metrics are introduced to analyze the visual similarity between sketches. The first metric is called *distance-based similarity*, which measures the distances among centroids of different sketch categories in the latent space. The shorter distance between two centroids means higher visual similarity between the two categories. The second metric is called *overlap-based similarity*, which measures the amount of overlap among cluster probability distributions of different sketch categories. The larger overlap between the two categories means higher visual similarity between the two. These two metrics work together to deal with different scenarios and provide more accurate measurements for visual similarity.

In *stage 6*, long- and short-distance analogies for each sketch category are identified based on visual similarity measures mentioned earlier. Sketch categories with high visual similarity are classified as short-distance visual analogies; otherwise, they are classified as long-distance visual analogies. A sketch category can easily build a visual relationship with short-distance categories. *Bridge categories* are identified to provide a way to discover valid long-distance visual analogies.

The proposed visual augmentation process is applied to sketches acquired from QuickDraw [64] as a case study. Section 4 presents detailed descriptions of the two main functions of the CAVAS framework.

4 Methods

4.1 Learn Shape Representations and Patterns With Deep Clustering. As mentioned earlier, a dimension reduction approach is taken to learn about the low-dimensional latent feature space of the given sketch datasets. More specifically, it is desired that a *generative model* can be trained that can discover embedded shape patterns of different sketch categories in the latent feature space without supervised information (e.g., category labels). Among various deep generative models for reconstructing images, VAE is one of the most widely used techniques because of its good performance in generalizing and learning a smooth latent representation of the input images.

Ha and Eck [58] proposed a sequence-to-sequence VAE for generating sketch drawings for completing a user's stroke-based drawing sequence of common objects. In this model, the stroke-based sketch drawings are captured as a RNN that can carry out conditional and unconditional sketch generation. Partly due to its stroke-based modeling approach; however, it has a key limitation, which is the insufficient quality of learning latent representations of sketches from *multiple* categories. The limitation made it inadequate for CAVAS, as visual relationships between multiple categories need to be learned.

To overcome the limitation of learning from single-category sketches, Chen et al. [62] replaced the RNN layers with CNN layers so that they can deal with pixel-based sketches (i.e., images). This change also removed the limitation of single-category sketch drawings and made it possible for CNN to learn from sketches of multiple categories and generate a wide variety of sketches based on the user's input.

Since the CAVAS framework considers visual analogies from multiple categories, our generative model must learn from sketches of multiple categories. Following Ref. [58], the CAVAS deep learning-based sketch generative model called the *CAVAS-DL* model is defined as follows.

4.1.1 Shape Feature Learning. Given n sketches $\boldsymbol{x} = \{x_i \epsilon X\}_{i=1}^n$, X is the data space (i.e., the space of all the sketches, represented as images), CAVAS-DL encoder $\boldsymbol{q}_\phi(\cdot)$ compresses $\boldsymbol{x}$ into n latent vector $\boldsymbol{z} = \boldsymbol{q}_\phi(\boldsymbol{x}) = \{z_i \epsilon Z\}_{i=1}^n$. Z is the *latent space*. The dimensionality of Z is typically much smaller (e.g., 128) than X (e.g., 2304).

CAVAS-DL decoder $\boldsymbol{p}_\theta(\cdot)$ samples n sketches conditional on $\boldsymbol{x}' = \boldsymbol{p}_\theta(z) = \{x_i' \epsilon X\}_{i=1}^n$ given latent vector z. The loss function of the model can be defined as follows:

$$L_r = E_{q_\phi(z|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}'|z)] \tag{1}$$

where ϕ and θ are the parameters to be trained in the encoder and decoder, respectively. The parameters are typically the weights and biases of the neural networks. $E_{q_\phi(z|\boldsymbol{x})}(\cdot)$ is the *reconstruction loss* that ensures the close resemblance between the generated sketches and the original sketches.

As shown in Fig. 3, the CAVAS-DL encoder $\boldsymbol{q}_\phi(\cdot)$ is implemented as a deep CNN that maps the black-and-white images in the space of $48 \times 48 = 2304$ dimensions into vectors in a latent space Z of 128-dimension. Because the encoder is modeled as a generative variational autoencoder, the vectors in Z capture the shape feature in terms of normal distributions of the shapes in the original space and take pairs of *mean* and *standard deviation* as values. The CAVAS-DL decoder $\boldsymbol{p}_\theta(\cdot)$ is modeled as an RNN that samples from the latent space Z and reconstructs the corresponding sketch images.

4.1.2 Embedded Clustering. To identify short- and long-distance analogies, sketches *sharing more shape features* should be grouped and separated from other groups. Clustering is an unsupervised learning method that can cluster similar data points into the same group. In ordinary situations, clustering of data points starts when the dimensional space of the data points is given and depends only on the settings of distance measures and clustering objectives. In the case of the CAVAS-DL model, however, clustering of sketches happens in the latent space Z that is being learned through training. The challenge here is how we can devise a clustering process that can not only perform the clustering task in Z but also help the training process of learning about Z and hence the mapping parameters of $\boldsymbol{q}_\phi(\cdot)$ and $\boldsymbol{p}_\theta(\cdot)$.

Xie et al. [65] proposed a deep embedded clustering (DEC) method to provide a way to simultaneously learn feature representations and clustering assignments using deep neural networks. This is especially difficult because of the nature of unsupervised learning in clustering. The key idea of DEC is to iteratively refine clusters with an auxiliary target distribution derived from the current soft cluster assignment between the data points and the cluster centroids. This process gradually improves the clustering as well as the feature representation.

The DEC method is adopted in CAVAS-DL for improved mapping and clustering. As shown in Fig. 3, the clustering layer clusters all vectors in the latent space Z by simultaneously learning a set of K cluster centers $\{\mu_j \epsilon Z\}_{j=1}^K$ and mapping each latent vector z_i into a soft label q_i by Student's t-distribution [66]. $\boldsymbol{q_i} = [q_{i1}, \ldots, q_{ij}, \ldots q_{ik}]$ is a *soft label*, which quantifies the similarity between z_i and cluster center μ_j.

$$q_{ij} = \frac{(1 + \|z_i - \mu_j\|^2)^{-1}}{\sum_j (1 + \|z_i - \mu_j\|^2)^{-1}} \tag{2}$$

where q_{ij} is the jth entry of $\boldsymbol{q_i}$, representing the probability of z_i belonging to cluster j.

The clustering loss L_c is defined as a KL divergence (often used to measure how one probability distribution is different from a reference distribution) between the distribution of soft labels Q measured by Student's t-distribution and the *predefined target* distribution P derived from Q. The clustering loss is defined as follows:

$$L_c = D_{KL}(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{3}$$

where the *target* distribution P is defined as follows:

$$p_{ij} = \frac{q_{ij}^2/f_j}{\sum_j (q_{ij}^2/f_j)} \tag{4}$$

Raising q_{ij} to the second power and then dividing by the frequency per cluster, $f_j = \sum_i q_{ij}$, allows the target distribution P to improve cluster purity and put emphasis on confident labels. At the same time, this target distribution normalizes the contribution of each centroid to the clustering loss to prevent large clusters from distorting the latent space. This iterative strategy to minimize L_c works like self-training that labels the dataset to train on its *high confidence* predictions [67].

The total loss function of CAVAS-DL, L_{rc}, is composed of two components: the reconstruction loss L_r in Eq. (1) and clustering loss L_c in Eq. (3). L_r is used to learn abstracted representations of the latent space in an unsupervised manner that can preserve shape features in sketch datasets. L_c is responsible for manipulating the latent space to cluster sketches based on shape similarity. The purpose of the loss function L_{rc} is to minimize reconstruction loss L_r and clustering loss L_r. A weighted sum method is used to optimize L_r and L_c:

$$L_{rc} = L_r + \tau L_c \tag{5}$$

where L_r is from Eq. (1) and L_c is from Eq. (3), and coefficient τ is set to be $0 \le \tau \le 1$.

Jiang and Luo [57] introduced an RNN-based stroke-based modeling approach. However, it has a key limitation, which is the insufficient quality of learning latent representations of sketches from *multiple* categories. Goldschmidt [63] replaced the RNN layers with CNN layers so that they can deal with images and learn from sketches of multiple categories. Our method takes CNN as the encoder layer and applies an embedded approach to carry out feature learning and clustering simultaneously. Its advantage over these previous methods has been demonstrated in Ref. [23].

4.1.3 Training. The shape feature mapping parameters ϕ and θ of CAVAS-DL are pretrained by setting $\tau = 0$ to establish an initial latent space. After pretraining, the cluster centers are initialized by performing k-means on latent features of all sketches to obtain initial cluster centers $\{\mu_j \epsilon Z\}_{j=1}^K$. Based on Eqs. (2) and (4), the initial distribution of soft labels Q and initial target distribution P can be obtained. Then, the deep clustering weights, cluster centroids, and target distribution P are updated as follows:

(1) Update *weights* and *cluster centroids*. The gradients of L_c for each latent vector z_i and each cluster center u_j can be computed as follows:

$$\frac{\partial L_c}{\partial z_i} = 2\sum_{j=1}^{k} (1 + \|z_i - \mu_j\|^2)^{-1}(p_{ij} - q_{ij})(z_i - \mu_j) \tag{6}$$

$$\frac{\partial L_c}{\partial u_j} = 2\sum_{i=1}^{n} (1 + \|z_i - \mu_j\|^2)^{-1}(q_{ij} - p_{ij})(z_i - \mu_j) \tag{7}$$

Encoder and decoder parameter gradient $\partial L_r/\partial \phi$ and $\partial L_r/\partial \theta$ can be calculated by backpropagation when passing $\partial L_c/\partial z_i$ to the structure of the CAVAS-DL model. Then, the *parameters* of encoder and decoder, ϕ and θ, and the *cluster center*, μ_j, can be simultaneously updated by mini-batch stochastic gradient descent.

(2) Update *target distribution*. In every epoch of training, the target distribution P serves as ground truth soft labels. The clustering layer is trained by predicting the soft assignment Q and then matching it to the target distribution P. At the end of the epoch, based on Eq. (4), the target distribution P is updated depending on the predicted soft label Q and used for the next epoch. After each epoch, the cluster label c_i assigned to z_i is obtained by

$$c_i = \arg\max_j q_{ij} \tag{8}$$

where q_{ij} can be obtained from Eq. (2). The training will stop when the cluster label assignment change (in percentage) between two consecutive epochs is less than a threshold δ.

4.2 Analyze Visual Similarity to Identify Visual Analogies. The output of the clustering layer is a probability distribution of each latent vector z_i into each soft clustering label j. A clustering space can be introduced by any one-dimensional vector $\boldsymbol{\rho} \in \mathbb{R}^l$ that represents a probability distribution of clustering. Therefore, $\boldsymbol{\rho} = [p(c_1|\rho), \ldots, p(c_k|\rho), \ldots, p(c_l|\rho)]$, $c_k (1 \le k \le l)$ represents the kth cluster with $p(c_k|\rho)$, indicating the probability of data ρ that belong to the kth cluster.

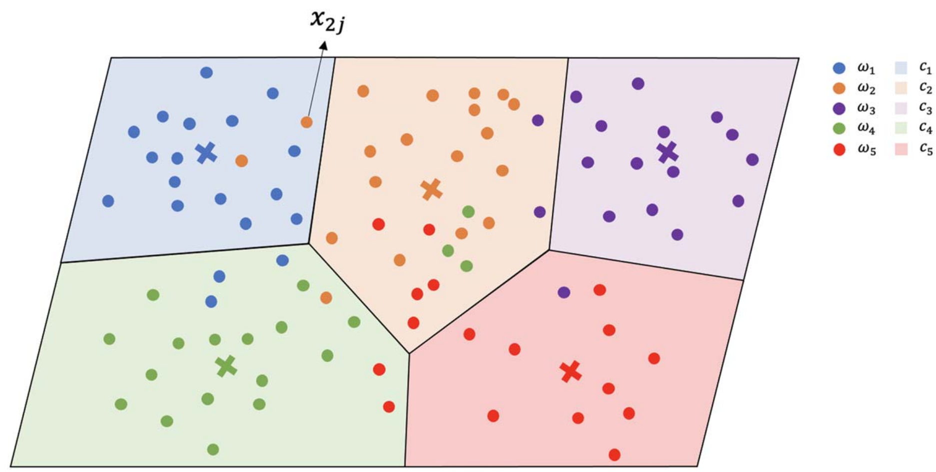


Fig. 4 Sketches from five categories in a clustered five-dimensional space

In CAVAS-DL, the inputs are sketches belonging to different categories, $\boldsymbol{x} = [x_{11}, \ldots, x_{ij}, \ldots, x_{st}]$, where x_{ij} means the jth sketch belonging to the ith category, s is the number of categories and t is the total number of sketches. In the latent space, latent vectors are $\boldsymbol{z} = [z_{11}, \ldots, z_{ij}, \ldots, z_{st}]$. In the clustering space, the probability distributions of latent vectors can be represented by a super matrix $\mathbb{Q}$, $\mathbb{Q} = [\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_s]$. For matrix $\boldsymbol{Q}_i (1 \leq i \leq s)$, it includes n sketches. $\boldsymbol{Q}_i = [\boldsymbol{q}_{i1}, \ldots, \boldsymbol{q}_{ij}, \ldots, \boldsymbol{q}_{in}]$, where $\boldsymbol{q}_{ij} (1 \leq j \leq n,\ n * s = t)$ represents a latent vector z_{ij} in the clustering space, i.e., $\boldsymbol{q_{ij}} = [p(\mathrm{c}_1|z_{ij}), \ldots, p(\mathrm{c}_k|z_{ij}), \ldots, p(\mathrm{c}_l|z_{ij})]$, where $P(\mathrm{c}_k|z_{ij})$ means the probability of z_{ij} belonging to the cluster c_k, $\sum_1^l P(\mathrm{c}_k|z_{ij}) = 1$. Soft clustering produces multiclustering predictions for x_{ij}, while the ground truth category of x_{ij} is single labeled.

Figure 4 illustrates a clustered five-dimensional space. Circle "o" indicates an input sketch, and cross "×" represents a centroid. The sketches of different categories are rendered with different colors. Solid lines indicate decision boundaries that are perpendicular bisectors of adjacent cluster centers, and the clusters are also rendered with different colors. As an example, in this five-dimensional clustering space, it is assumed that all sketches come from five categories, ω_1, ω_2, ω_3, ω_4, and ω_5. Given the ground truth category of x_{2j} is ω_2, the probability distribution of corresponding latent vector z_{ij} may be $\boldsymbol{q_{2j}} = [0.32, 0.21, 0.12, 0.19, 0.16]$. The cluster prediction of z_{ij} is c_1, which has a maximum probability of 0.32. However, sketches are clustered based on shape similarity. Sketches from different categories can be clustered in the same group. Hence, the concept of the *sketch category*, which often indicates what a sketch is in the real world, is different from that of the *sketch group* (or *shape cluster*), which clusters sketches based on their shape similarities.

4.2.1 Sketch Category and Sketch Group. We assume that the number of clusters equals the number of sketch categories, and each cluster can represent one *shape pattern* that is composed of several *shape features*. As shown in Fig. 4, there are sketches from the categories ω_1, ω_2, ω_3, ω_4, and ω_5, and there are five clusters c_1, c_2, c_3, c_4, and c_5. Each cluster can include sketches from several categories. For example, cluster c_2 contains sketches from the categories ω_2, ω_3, ω_4, and ω_5. It means that each cluster presents a *shape pattern* that is obtained from learning *shape features* from multiple categories. In other words, different categories can share one common shape pattern.

Sketches from the same category can be clustered into different groups. For example, some sketches in the category ω_5 are clustered into the clusters c_2, c_4, and c_5. It means that this category contains various shape features, which are learned by the CAVAS-DL to form different clusters, i.e., shape patterns. For a given category, the cluster label of each sketch is determined by Eq. (8), and the number of sketches in each cluster can be counted. The probability of category i belonging to cluster k is o_{ik}, which indicates the ratio of how many sketches in category i belong to cluster k and can be computed in Eq. (9). The cluster probability distribution of each category is represented by $O_i = [o_{i1}, \ldots, o_{ik}, \ldots, o_{il}]$.

$$o_{ik} = \frac{n_{ik}}{N_i} \tag{9}$$

where n_{ik} is the number of sketches from category i, which are located in cluster k, N_i is the total number of sketches in category i, and l is the total number of clusters.

4.2.2 Similarity Metrics. In this article, the first visual similarity metric is a *distance-based similarity* that measures visual similarity based on the Euclidean distance between the category centroids in the latent space. The centroid of a category can be obtained by averaging all the latent vectors from the same category. A category centroid is different from a cluster centroid, which is the centroid of all sketches (maybe from different categories) clustered in the same group. The distance-based similarity between category i and other categories can be computed as follows:

$$S_{ij}^{e} = 1 - \frac{E_{ij}}{\max\limits_{j} E_{ij}} \tag{10}$$

where E_{ij} is the Euclidean distance between the centroids of category i and j, and $\max\limits_{j} E_{ij}$ is the longest Euclidean distance from the centroid of category i to centroids of other categories.

The second metric is an *overlap-based similarity* that measures visual similarity based on the amount of shape feature overlap between sketch categories. Shape feature overlap is defined as the amount of overlap between two cluster probability distributions. If two categories share more shape features, their sketches are more likely clustered into the same groups. In other words, their probability distributions are closer and have more overlapping regions. Hellinger distance is applied to measure the similarity of two cluster probability distributions, which is defined as follows:

$$H(O_i, O_j) = \sqrt{1 - \sum_{k=1}^{l} \sqrt{o_{ik} o_{jk}}} \tag{11}$$

where $\sum_{k=1}^{l} \sqrt{o_{ik} o_{jk}}$ is a measure of the area intersected by two cluster probability distributions.

The overlap-based similarity of other categories to category i can be defined as follows:

$$S_{ij}^{o} = 1 - \frac{H(O_i, O_j)}{\max_j H(O_i, O_j)} \tag{12}$$

where $\max_j H(O_i, O_j)$ is the longest Hellinger distance from category i to other categories.

4.2.3 Short-Distance and Long-Distance Visual Analogies. Based on the aforementioned metrics, it is conceivable that the categories having high visual or shape similarity are likely to be clustered in the same group as their similarity values are all above a given threshold. In this research, sketch categories in the *same group* are considered visually *short distanced*. The value of the similarity threshold determines how "short" the distance must be for two categories to be considered short distanced. From a visual analogy support point of view, given a designer is working on sketching in category a and categories a and b are short-distanced, the CAVAS-DL may provide a sketch of category b as a visual cue to stimulate the designer's visual analogy thinking. In this case, the visual analogies made by the designer are likely to be *short-distance* ones. On the other hand, if categories a and b belong to *different groups*, then the analogies are likely to be *long-distanced* ones.

Identifying long-distance visual cues requires relating sketch categories belonging to different groups, which can be time consuming when the number of sketches and the number of categories are both large. To deal with this issue, a concept of *bridge category* is introduced. If there is a bridge category existing between two groups, the visual relationships between the categories in these groups can be established.

In Fig. 5, the solid dots are categories that are clustered into two groups. The similarity of category a to category b can be represented by the similarity value S_{ab}^{o} or S_{ab}^{e}. The similarity of category b to category a can be represented by the similarity value S_{ba}^{o} or S_{ba}^{e}. If S_{ab}^{o}, S_{ab}^{e}, S_{ba}^{o}, and S_{ba}^{e} are all equal to or greater than a threshold ε, category a and category b can be classified in the same group and become *short-distance* analogies.

For category b from group 1, category c is the closest category in group 2, and for category c, category b is the closest category in group 1. The similarity of categories b to c can be represented by the similarity values S_{bc}^{o} and S_{bc}^{e}. Category b is defined as a *bridge category,* if and only if S_{bc}^{o} or S_{bc}^{e} is equal to or greater than a threshold φ. In this case, there exists a visual relationship between categories b and c. As categories in group 1 are visually similar to category b and category b is visually similar to category c, other categories in group 1 can be visually similar to category c and then potentially visually similar to other categories, say category d, in group 2. If a bridge category is identified, it is possible to transfer the shapes of categories between these groups based on visual similarities. The process of finding a valid *long-distance* visual analogy is expressed as follows:

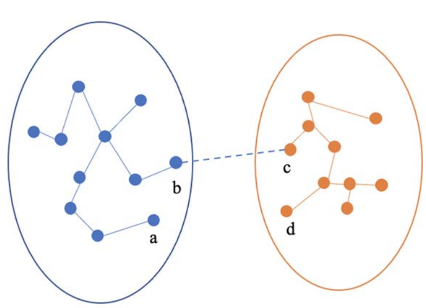


Fig. 5 Visual relationships between two groups of categories

$$\text{Given } a, b \in S \,\&\, c, d \in T; \text{ if } \; b \sim c, \text{ then } a \approx d$$

where S is a source domain of categories and T is a target domain of categories; $b \sim c$ means a visual relationship built between categories b and c; and $a \approx d$ means a possible long-distance visual relationship between categories a and d.

5 Experiments

5.1 Datasets and Implementation. Currently, there are few large engineering design image datasets available for us to train our model. The CAVAS-DL model is evaluated based on the image datasets from QuickDraw, the largest sketch database built by Google [64] to date. The proposed methods are for any generic sketch-based visual analogies, as nothing in the dataset specifically ties the work to engineering problems. QuickDraw contains 345 categories of everyday objects. To consider the burden of computing time, sketches from ten categories are chosen to test our proposed methods. The raw sequences from QuickDraw datasets are converted to monochrome png files of size 48×48, which are used as the input data for our deep neural network. These png files are binary images with pixels covered by strokes having the value 1 and the rest of the pixels having the value 0. Three datasets from ten categories are used for the experiments:

Dataset 1: Includes five categories, which are *van*, *bus*, *truck*, *pickup truck*, and *car*. All of them belong to automobiles and share some obvious shape features such as wheels and windows.

Dataset 2: Includes five categories, which are *speedboat*, *canoe*, *drill*, *pickup truck,* and *car*. Speedboat and canoe belong to boats and share some obvious shape features such as V-shaped hulls. *Pickup truck* and *car* belong to automobiles. *Drill* does not share superficial shape similarities with other categories.

Dataset 3: Includes five categories, which are *television*, *canoe*, *drill*, *umbrella*, and *car*. Each of them does not share any superficial shape similarities with other categories.

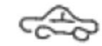
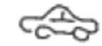
Some examples of each dataset are listed in Table 1. The 15K sketches for each category are chosen. The sketches are divided into *training*, *validation*, and *testing* sets with sizes of 10 K, 2.5 K, and 2.5 K, respectively.

To quantitatively verify and demonstrate the improved performance of CAVAS-DL, a comparison study between Cava-DL and the work of *sketch-pix2seq* proposed by Chen et al. [62] and its predecessor *sketch-rnn* by Ha and Eck [58] was conducted. For the sake of completeness, one of the traditional clustering algorithms, *k-means* is also included in the comparison.

The experiments on the four methods, namely, *CAVAS-DL*, *sketch-pix2seq + k-mean, sketch-rnn + k-mean*, and *k-mean*, are conducted using the three datasets described earlier. The parameters used for training *sketch-rnn* and *sketch-pix2seq* models are the same as the illustration in the papers [58,62]. CAVAS-DL is initialized by pretraining with $\tau = 0$, i.e., with the deep clustering detached. Then, the coefficient τ of clustering loss in Eq. (5) is set to 0.05, which is determined by a grid search in a list [0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0] and batch size to 100 for all datasets. The maximum number of epochs is set to $T = 50$. In each iteration, we train the encoder for one epoch using Adam optimizer with a learning rate $\lambda = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. The convergence threshold δ is set to 0.1%. The dimension of the latent space in these three models is 128, which is the same in the papers [58,62]. k-Means is performed to cluster sketches in the latent space of sketch-pix2seq and sketch-rnn. Besides, as a baseline for comparison, k-means also runs on the sketch datasets with the original dimensions of $48 \times 48 = 2304$, which is much larger than the latent space. k-Means performs 20 times with different initialization, and the result with the best objective value is chosen, where $k = 5$.

We evaluate all four clustering methods with unsupervised clustering accuracy (ACC). The ACC is defined as the best match

Table 1 Examples of each dataset

Dataset	Examples				
1	 Van	 Bus	 Truck	 Pickup truck	 Car
2	 Speedboat	 Canoe	 Drill	 Pickup truck	 Car
3	 Television	 Canoe	 Drill	 Umbrella	 Car

between ground truth $\boldsymbol{y}$ and predicted cluster labels $\boldsymbol{c}$:

$$ACC(\boldsymbol{y}, \boldsymbol{c}) = \max_{m \in \mathcal{M}} \frac{\sum_{i=} \mathbf{1}^n 1\{y_i = m(c_i)\}}{n} \tag{13}$$

where n is the total number of samples, y_i is the ground truth label, c_i is the predicted cluster label of example x_i obtained by the model, and $\mathcal{M}$ is the set of all possible one-to-one mappings between predicted cluster labels to ground truth cluster. The best cluster assignment can be efficiently computed by the Hungarian algorithm [68].

5.2 Shape Feature Learning and Clustering Performance. As described in Sec. 3.1, to provide adequate visual cues to stimulate the designer's analogical thinking, the CAVAS system should be able to learn the shape features from the given datasets and distinguish the shape patterns that go beyond the sketch categories. From feature learning and clustering perspectives, the major distinction of our proposed CAVAS-DL method is combining deep feature learning with deep clustering. Thanks to the dynamic property of CAVAS-DL that simultaneously adjusts the processes of feature learning and clustering, its improved performance in shape pattern identification is expected, and in fact, it has been reported that our proposed algorithm outperforms others significantly for both clustering and category information preservation [23].

For visualizing the latent space of unsupervised modes and one supervised model on the three datasets, t-SNE [66] is used to reduce the dimensionality of Z from 128 to 2, and 7500 testing sketches are plotted from five categories of the three datasets for each method. The dimensionality reduction from 128 to 2 may cause significant information loss and generate misleading visualizations. t-SNE has a hyperparameter called perplexity. The perplexity value balances the attention t-SNE gives to local and global aspects of the data and can have large effects on the resulting plot. It is recommended to be between 5 and 50 [69]. If choosing different values between 5 and 50 significantly changes the interpretation of the data, then t-SNE is not the best choice to visualize or validate our hypothesis. To increase the robustness of our findings and reflect how multiple runs reflect affect the outcome of t-SNE, we put forward the process to validate the visualization of a trained latent space, which is shown in Fig. 6.

In the first step, we set the initial value of the counter N as 0, which is used to record the times of sample generation. Then, we take advantage of t-SNE for visualizing a latent space with a list of perplexity values. In the second step, we choose a converged visualization as the candidate. For example, in Fig. 7, the latent

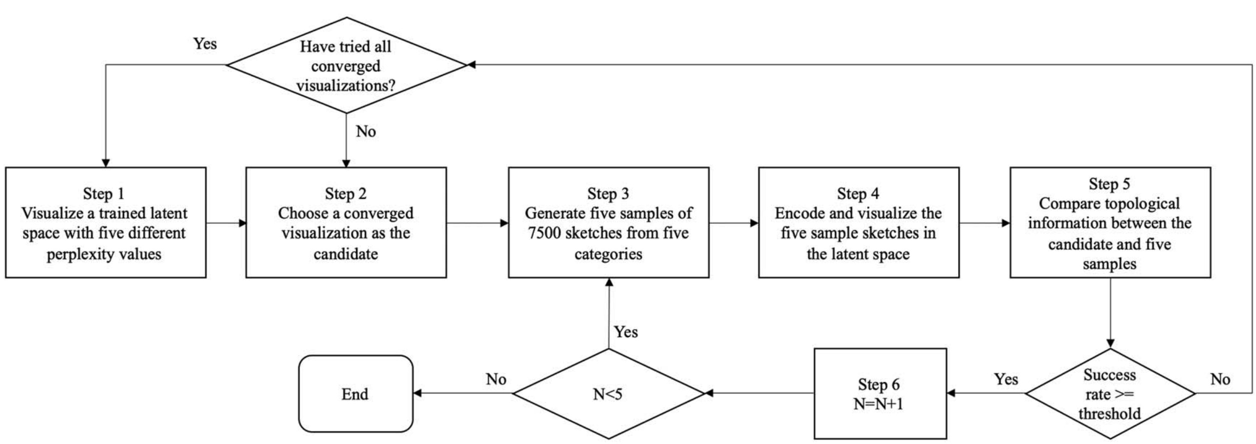


Fig. 6 The process to validate the visualization of a trained latent space

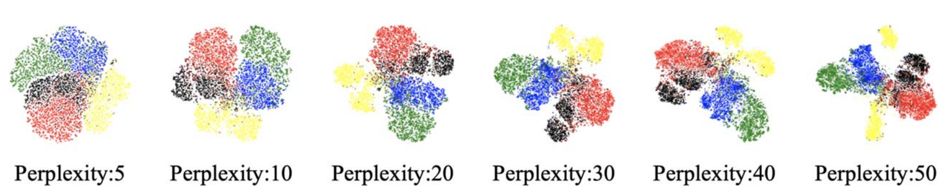


Fig. 7 Visualizations of a latent space with different perplexity values

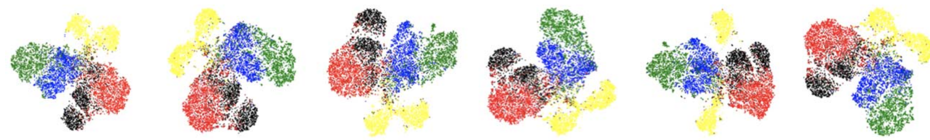


Fig. 8 Comparison of visualizations of five sample sketches in the latent space with the candidate

space is visualized under different perplexity value settings. We can see the latent space visualizations in a list [30,40,52] are converged. There are two types of global geometry of converged visualizations. One type can represent visualizations with perplexity values of 20, 30, and 40. Another type can represent the visualization with a perplexity value of 50. We randomly choose one (perplexity value = 30) as the candidate from the first type. If this type cannot be a valid visualization, we will try the other types. In the third step, we randomly generate five samples of 7500 sketches from five sketch categories in the QuickDraw dataset. In the fourth step, the five sample sketches are encoded and visualized in the latent space. In the fifth step, we compare the topological information of five samples in the latent space with the candidate. If over half of the visualizations of the five samples are similar to the candidate, it means this round of sample generation can validate that the candidate can represent these five sample sketches in the latent space. A success rate is used to illustrate how many samples are similar to the candidate. For example, in Fig. 8, only sample_4 is different from the candidate, and the success rate is 0.8. A threshold is set to 0.6. If the success rate is no less than a threshold, then the next round of five sample generations will be started. The counter N will be increased by 1. Otherwise, we will go back to the second step to choose the other type of converged visualization as the candidate. If all the converged visualizations have been tried, then we go back to the first step. There will be five rounds of sample generation. If all of them can be successful, then the candidate will be chosen to visualize the trained latent space.

After validating visualizations of the latent space of each model in three datasets, we can compare the shape feature learning and clustering performance of different models. First, we compare the learning and performance of unsupervised models. Figure 9 shows that the CAVAS-DL performs the best in clustering since the sketches from different categories are more separated than those from the same ones, and the sketches from the same category are denser together in all cases. For Dataset1, all sketches are from the same taxonomic category and hence are hard to be separated into different clusters. The red, black, and green clusters are denser in CAVAS-DL than the other two as the clustering loss L_c can force sketches from the same taxonomy to be gathered together and push away sketches from different taxonomies. It means high visually similar categories will gather together in the latent space, and low visually similar categories will separate from each other. This capability is helpful in identifying short- and long-distance analogies. For Dataset2, sketches are from three taxonomic categories. Sketch categories belonging to the same taxonomy should be close to each other as they share more shape features and are away from other taxonomies. This assumption can be confirmed by our proposed model as well as sketch-pix2seq, as they both use CNN as an encoder that can discover and represent shape structures in the latent space. Car (red) is close to pickup truck (black) and speedboat (blue) is close to canoe (green) in the first CAVAS-DL plot, while this cannot be easily detected in the third sketch-rnn plot; for Dataset3, all sketches are from different taxonomic categories. All three deep learning models can easily cluster each category. However, the clusters in the CAVAS-DL plot are denser and have a larger margin with each other. Besides the qualitative comparisons based on Fig. 9, we gave the quantitative results of the clustering performance of our model and other compared models in our previous paper [23]. The main conclusion is our model has a better performance in clustering with shorted intracluster distance and longer intercluster distance when using different datasets.

In Fig. 9, we also compare three unsupervised models mentioned earlier with a supervised model, which is a CNN from the official guides of QuickDraw.[2,3] For every dataset, the supervised model can more clearly separate each category in the latent space. The reason is the latent space of the supervised model is trained based on given category label information. Therefore, the supervised model can have better performance in categorizing sketches. From dataset1 to dataset3, the shape feature sharing become less and less, and the margins between sketch categories in the latent space become larger and larger. It infers that after training, shape features extracted by the encoder are related to the given semantic information (category labels). When all sketches are from the same taxonomy, this relationship can be hardly built. When all sketches are from different taxonomies, this relationship can be easily established. However, in this research, the goal is to learn a latent space that represents the shape patterns. Ideally, similar shapes from the same or different categories can be clustered in the same group, and different groups are distinguishable from each other. In other words, the purpose of the proposed CAVAS-DL is to construct a relationship between shape features and shape patterns in case the shape pattern label of each sketch is hard or impossible to be collected and created. Therefore, even though all sketch categories are from different taxonomies in dataset3, CAVAS-DL tries to keep relatively small margins to possibly build shape connections between these categories.

5.3 Performance of Visual Similarity Analysis. After extracting shape features and discovering shape patterns from the given datasets, the CAVAS system should be able to analyze visual similarities between different sketch categories and identify relevant visual cues. To measure visual similarity, both *distance*- and *overlap*-based *similarities* are introduced.

Euclidean distance. t-SNE always uses the Euclidean distance function to measure distances because it is the default parameter set inside the method definition [66]. One limitation of using Euclidean distance in t-SNE is the quadratic time and space complexity in the number of data points as it computes all pairwise similarities between the points. One good practice is to first use principal component analysis to reduce the number of dimensions and then use t-SNE [70,71]. However, we do not take this approach as only ten categories are chosen for the testing. In this article, we choose the Euclidean distance as the distance-based similarity measurement. In Fig. 10, the clustered latent space is presented to visually show Euclidean distances between centroids of ten sketch categories. Speedboat and canoe in the green circle are from the same taxonomy, and van, pickup truck, truck, car, and bus in the red circle are also from the same taxonomy. Pickup truck and speedboat are close to each other, and hence, it is possible to build a visual relationship between two taxonomies through these two categories. Drill, television, and umbrella are from different taxonomies.

[2]https://github.com/googlecreativelab/quickdraw-dataset
[3]https://github.com/zaidalyafeai/zaidalyafeai.github.io/tree/master/sketcher

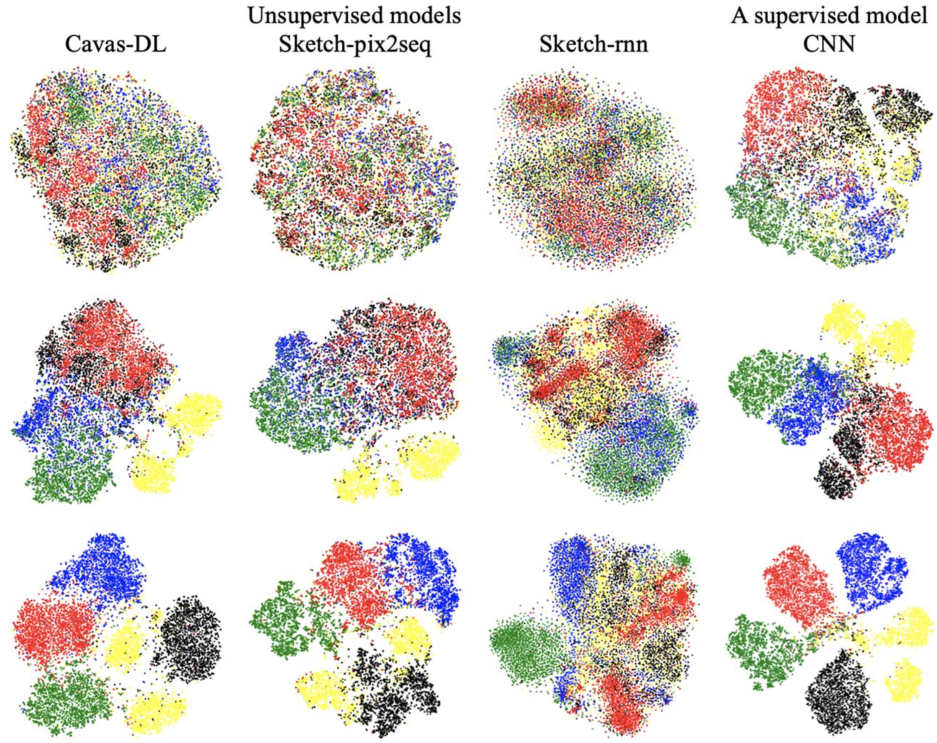


Fig. 9 Clustered latent space of three datasets for each method (top row: Dataset1—van (blue), bus (green), truck (yellow), pickup truck (black), car (red); middle row: Dataset2—speedboat (blue), canoe (green), drill (yellow), pickup truck (black), car (red); bottom row: Dataset3—television (blue), canoe (green), drill (yellow), umbrella (black), car (red)) (Color version online.)

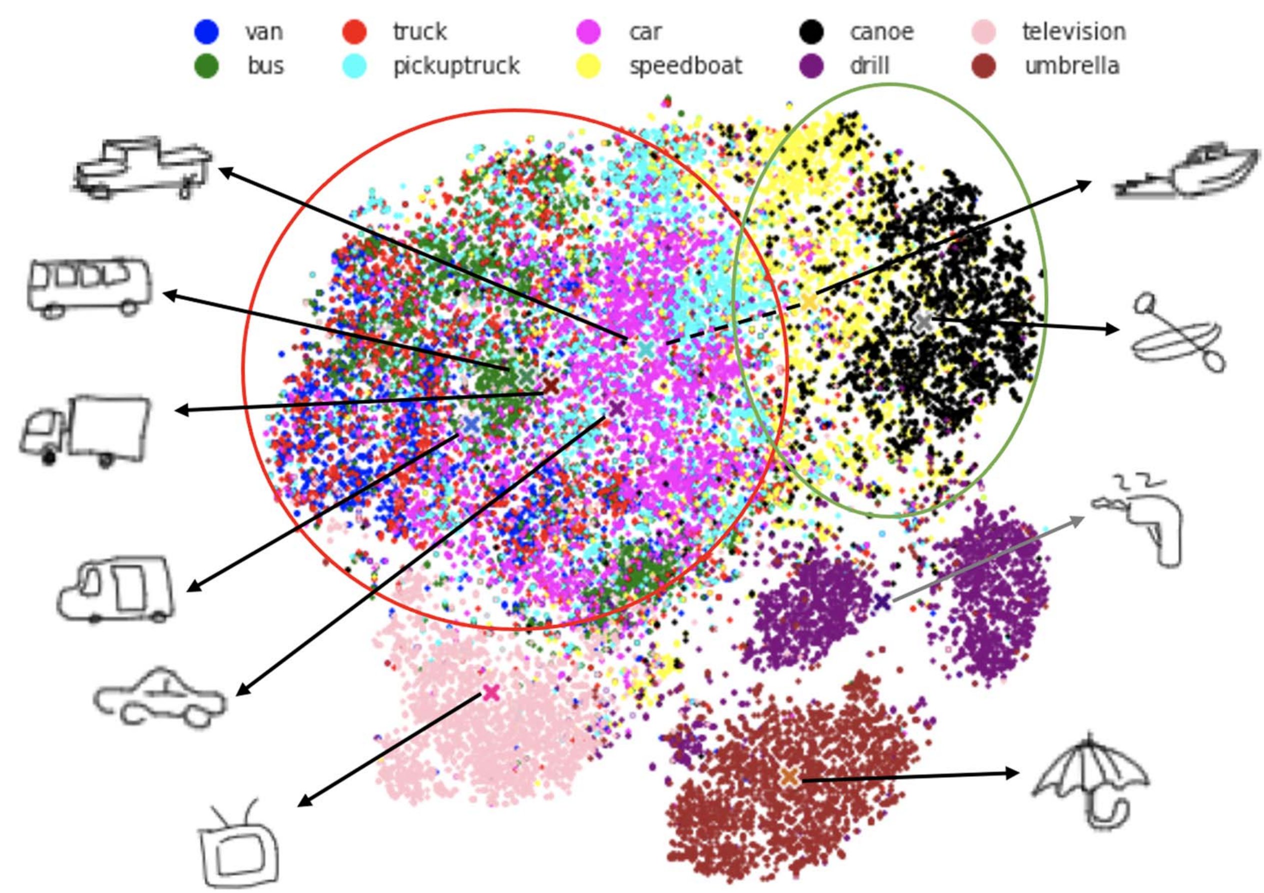


Fig. 10 Sketches from ten categories in the latent space, cross "×" represents a category centroid (Color version online.)

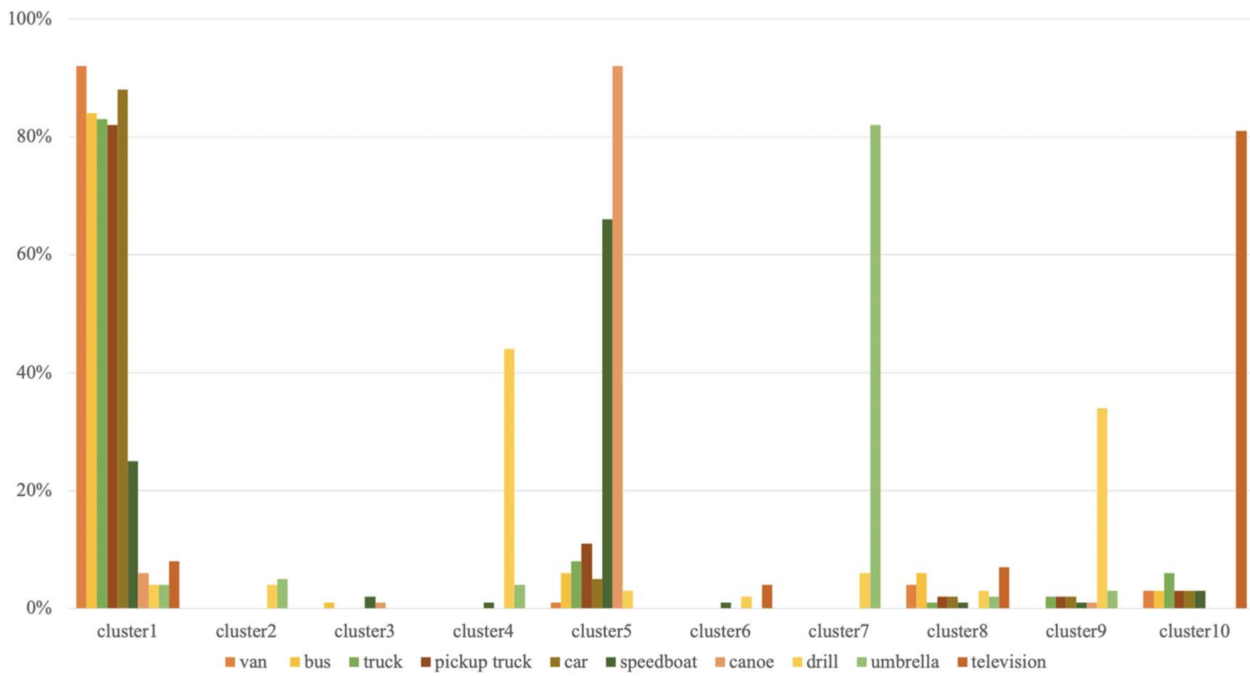


Fig. 11 Cluster probability distributions of ten categories

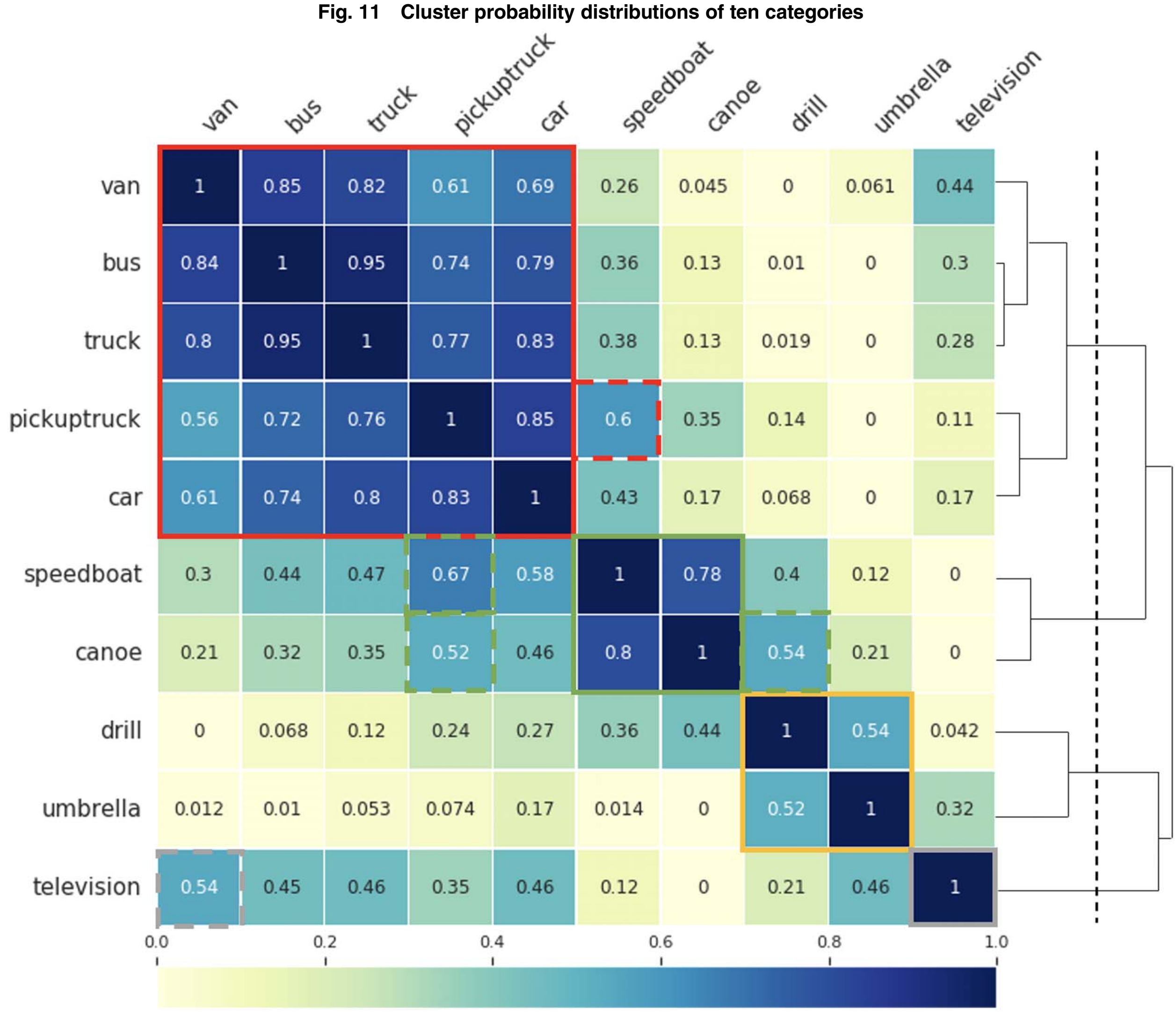


Fig. 12 A distance-based similarity matrix with dendrograms (different groups are marked with solid squares with different colors; some cells' values larger than threshold φ are marked with dashed squares to indicate bridge categories) (Color version online.)

Categories from the same taxonomy have shorter centroid distances and higher overlap magnitude; Categories from different taxonomies have longer centroid distances and lower overlap magnitude. After checking the dataset, one could see that there are two different kinds of drills in the dataset: *handheld drill* and *ground drill*. Therefore, drills are separated into two groups in the latent space. By exploring this latent space of ten categories, designers can have an overall view of the visual relationships between them.

Hellinger distance. The cluster probability distributions in Fig. 11 are provided to visually present the amount of overlap between categories based on Hellinger distance. The following features are shown in Fig. 11**.**

(1) *A cluster can accurately capture one shape pattern*, which can represent shape features from the same or different taxonomies. For example, cluster 1 captures most shape features from the automobile taxonomy. It can also capture some shape features from the speedboat; cluster 5 captures most shape features from the boat taxonomy. It can also capture some shape features from pickup truck. This also explains why speedboat and pickup truck can be *bridge categories* to link boat and automobile taxonomies. Cluster 7 and cluster 10 capture shape features from umbrella and television, respectively. Umbrella and television are distinguishable from other categories.
(2) *One category might contain two or more shape patterns as it has different variations*. This case can be captured by different clusters. For example, cluster 4 and cluster 9 capture different shape patterns in the *drill* because there are two different types of drills in the dataset.
(3) *Some categories may be extra or not useful*. Cluster 2, 3, 6, and 8 can barely capture some shape features from ten categories. It was assumed that the clustering layer can learn and differentiate shape features from different categories, and one category can only represent one shape pattern. Therefore, the number of clusters is set equal to the number of categories during the experiment. However, the results showed that some clusters are not useful. Therefore, the optimal cluster number needs to be determined by iterating more experiments. We believe the optimal cluster number may relate to the number of taxonomies in the given dataset, as categories that belong to the same taxonomy would have the same shape pattern.

Visual similarity. The distance- and overlap-based similarity matrices in Ref. [72] and Fig. 14 can quantify the visual similarity between each category based on Euclidean distance and Hellinger distance, respectively. As all distances from other categories to a given category are normalized by the maximum distance, these two matrices are asymmetric. The matrices capture the following useful information:

(1) *Hierarchical clustering*: The rows in these matrices are rearranged based on hierarchical clustering and accompanied by dendrograms describing the hierarchical cluster structure.
(2) *Similarity magnitude:* The values in each cell represent the *similarity magnitude* of the row category to each column category. A larger value means higher similarity.
(3) *Category groups*: If similarity values between several categories are all equal to or greater than the threshold $\varepsilon = 0.5$, then these categories can form a group (i.e., cluster or shape pattern).
(4) *Analogy types*: Categories in the same group are *short-distance* visual analogies. Categories in the different groups are *long-distance* visual analogies. The threshold φ is set to 0.5. A category can be considered as a *bridge category* if the largest similarity value between this category with one category in another group is equal to or greater than 0.5.

n Ref. [72], as threshold ε is set to 0.5, ten categories can form four groups based on distance-based similarity, which is shown in the dendrogram. *Van, bus, truck, pickup truck* and *car* are in the red group. *Bus* and *truck* have the highest similarity values. It implies they are tightly closed to each other in the latent space. The green group includes *speedboat* and *canoe*. The orange group contains *drill* and *umbrella*, which are from different taxonomies. The gray group contains *television*. The red group is entwined with the green group. It means shape transformation can happen between automobiles and boats as they share many shape features. The similarity value of *pickup truck* to *speedboat* is 0.6, and the value of *speedboat* to *pickup truck* is 0.67, which are above the threshold φ. They are bridge categories with a strong capability to connect two taxonomies. It means for making a visual analogy, if the target domain is *boat*, a boat designer can try to make a visual connection with a source domain which is automobile through *speedboat*, vice versa. As shown in Fig. 13, *van, bus, truck, pickup truck*, and *car* are different categories in the *automobile* group. It is more effective to build visual connections between them, but fewer changes to obtain visual inspiration. More efforts need to be made to construct a visual relationship between different categories in different groups (e.g., van and canoe). However, novelty is more likely to happen if a long-distance visual connection can be built. Bridge categories (*pickup truck* and *speedboat*) are valuable spots to draw a visual analogy for both effectiveness and novelty at the same time. *Canoe* is the only category that can connect one group with the other two

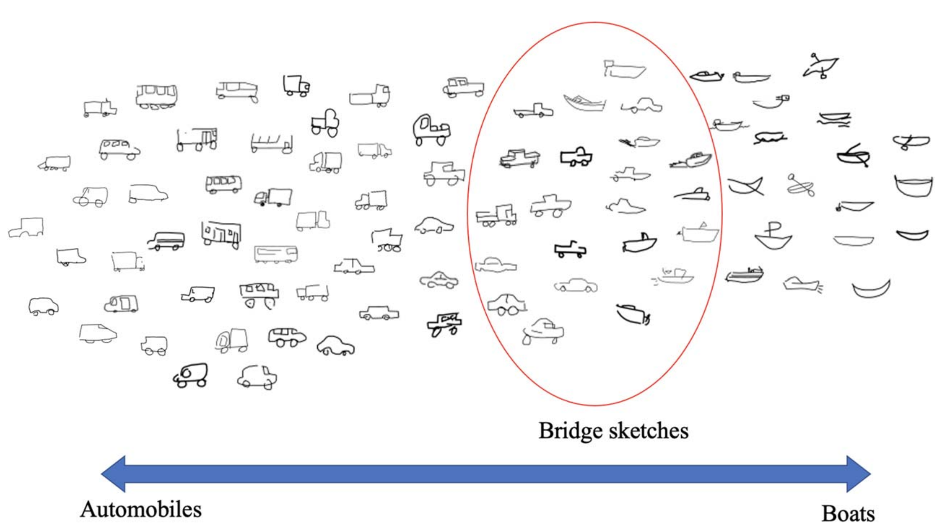


Fig. 13 A possible visual analogy making through bridge categories

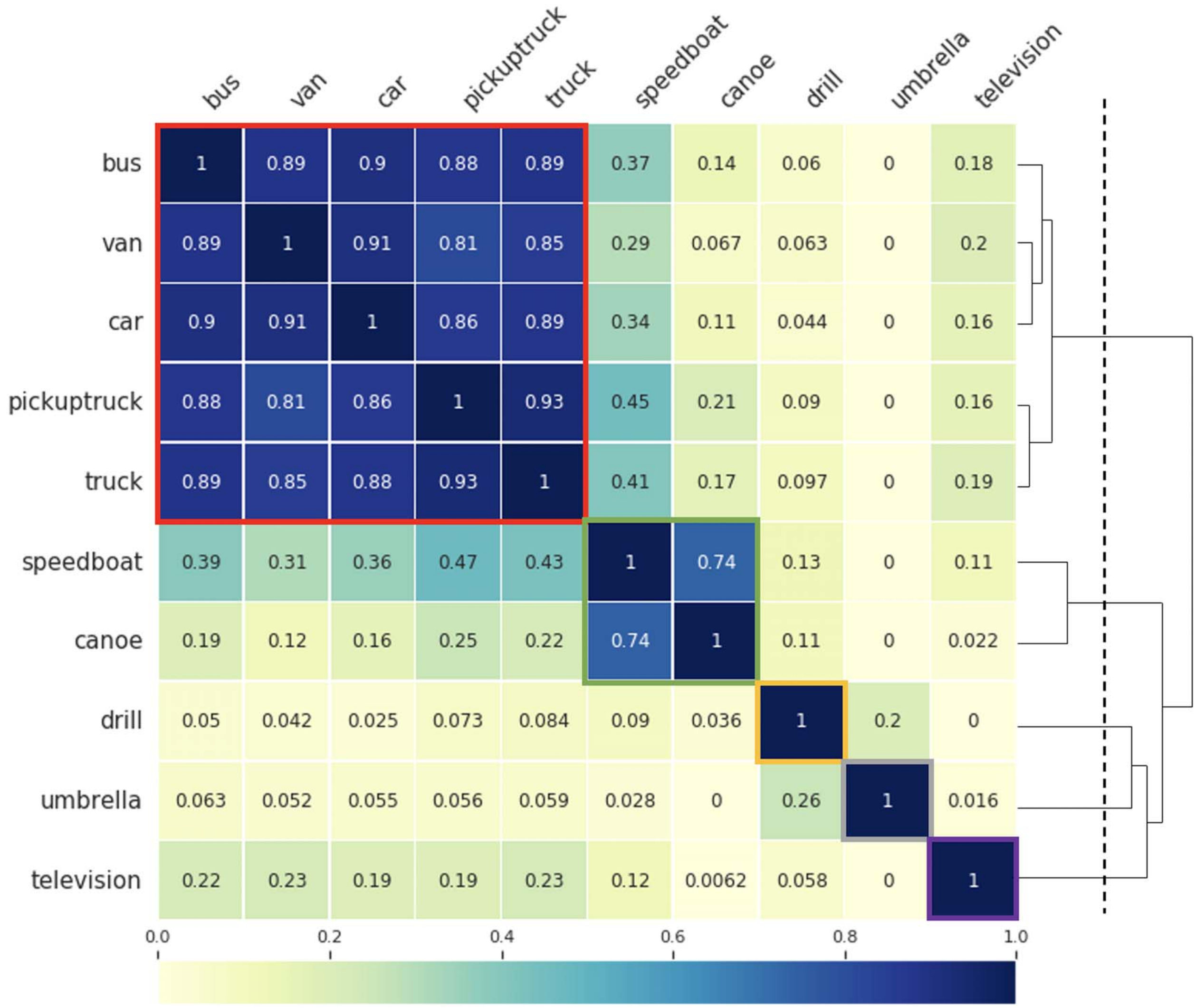


Fig. 14 An overlap-based similarity matrix with dendrograms (different groups are marked with solid squares with different colors) (Color version online.)

groups as the similarity values of canoe to *pickup truck* and *drill* are 0.52 and 0.54, respectively. It means it can lead visual connections to different directions. The similarity value of television to *van* in the red group is 0.54, which is above the threshold φ. It means *television* has a potential to make a visual relationship with *automobile*. It is easy to understand as the screen of the television is visually similar to a window of a van.

In Fig. 14, as threshold ε is set to 0.5, ten categories can form five groups based on overlap-based similarity, which is shown in the dendrogram. The categories in red and green groups are still the same. However, drill and umbrella are not classified in the same group. Basically, the similarity values between categories from the same taxonomy become larger, and the similarity values between categories for different taxonomy become smaller. For example, the lowest similarity value in the red group is increased by 0.25. Pickup truck and truck have the highest overlap-based similarity. It makes more sense compared with distance-based similarity from which truck is more visually similar to *bus*. No bridge categories can be detected. Therefore, the visual relationships between categories in the same group become stronger. However, each group is distinguishable from other groups. It may be hard to build visual relationships between these groups based on overlap-based similarity. In other words, short-distance visual analogies can be easily identified, but long-distance visual analogies might be harder to be found.

6 Discussion

Human designers are sophisticated in extracting essential visual features from shapes and discovering visual patterns to aid them in inferring analogies from different shapes. To learn shape patterns from sketches, our proposed CAVAS-DL takes advantage of CNN as the encoder to compress high-dimensional sketch image data from multicategory to low-dimensional features in the latent space. By minimizing the reconstruction loss L_r, our model can reduce shape information lost during compression and capture shape features in the sketch data. By minimizing the clustering loss L_c, sketches from the same category are densely clustered and away from other categories. It means sketches belonging to the same shape patterns are more likely clustered in the same group. These properties are proven in the experiments.

By visualizing the latent space of three different datasets with different levels of common shape feature sharing in Fig. 9, we empirically validate two points: (1) If sketches are from the same taxonomy, they share many shape features. It is difficult for deep clustering models to separate them. The sketches in Dataset1 are from the same taxonomy; three unsupervised models are struggling to cluster sketches. But our proposed model can somehow separate red and black points from others. The sketches in Dataset3 are from different taxonomies; it is easier for the three unsupervised models to cluster sketches. Our proposed model can separate clusters with a larger margin. (2) If a deep clustering model uses CNN layers to encode input sketches and simultaneously considers the clustering loss, its clustering performance can be improved, indicating the advantage of embedded clustering. Some of the sketches in Dataset2 come from the same taxonomy; our model can cluster points denser than the other two unsupervised models. (3) Among three unsupervised models, CAVAS-DL is the most similar to CNN regarding the sketch distributions in the latent space. It also suggests that CAVAS-DL is better at differentiating sketches based on shape patterns and also retaining shape relationships between sketches in the same taxonomy.

The scalability and repeatability are important aspects we need to explore for our model. We chose ten categories to test the performance of shape feature extraction and shape pattern learning of the proposed model. For shape feature extraction, the assumption is when the categories come from the same taxonomy, it would be hard for our model to extract shape features as these categories are visually similar to each other; when the categories come from different taxonomies, it would be easy for our model to extract shape features as these categories are not visually similar to each other. For shape pattern learning, the assumption is one learned shape pattern can be represented by one cluster, which can include several categories sharing similar shape features. Categories in the same cluster can be potentially regarded as short-distance analogies, and categories in different clusters can be potentially regarded as long-distance analogies. Categories from the same taxonomy group are more likely located in the same cluster. However, one shape pattern/cluster is not equal to one taxonomy. The main reason is some categories from different taxonomies share many similar shape features. For example, car and boat may be clustered into the same group. On the basis of these assumptions, we collected images from the same taxonomy group to form dataset1 (van, bus, truck, pickup truck, and car), from three different taxonomy groups to form dataset2 (speedboat, canoe, drill, pickup truck, and car), and from five different taxonomy groups to form dataset3 (television, canoe, drill, umbrella, and car).

Currently, we have tested our model on a relatively small-scale dataset and our model has good performance in shape feature extraction and shape pattern learning. For real cases, we need to train our model on hundreds or thousands of categories to find more short- and long-distance visual analogies. More modifications of the model and experiments need to be done when the size of the dataset is scaled up. However, we can see the potential of our model to identify and provide visual cues for visual analogy making in conceptual design.

After effectively encoding the source of analogies, potential targets need to be identified. During the visual analogy search process, designers qualitatively assess the similarity between visual materials. The moment to identify a bridge to connect or transfer one shape to another is often random and unpredictable. To quantify visual similarity, distance- and overlap-based similarity are introduced to analyze the visual relationships between categories and find useful analogies. Bridge categories are defined to guide the connection building of different shapes.

From the experiment of visual similarity analysis, one can see that (1) the distance- and overlap-based similarity metrics can confirm that categories from the same taxonomy share more shape features and have higher visual similarity than categories from different taxonomies; (2) distance-based similarity is less accurate than overlap-based similarity when finding visual relationships between categories from the same taxonomy as these categories share too many shape features, and in these cases, the overlap-based similarity is more effective than distance-based similarity; (3) overlap-based similarity can make categories from different taxonomies more distinguishable, e.g., the visual similarity values between categories in the automobile taxonomy become larger. However, finding bridge categories become more difficult, e.g., the similarity values between speedboat with other categories in the automobile taxonomy become smaller, and it is not detected as a bridge category; (4) bridge categories can be useful to find the path to visually transform shapes from one taxonomy to another taxonomy. The path can potentially explain how to find long-distance visual analogies. For example, *pickup truck* is classified as a bridge category. A car designer can apply visual thinking to transfer the shape of a car to a pickup truck and then to a speedboat and retrieve some inspiring cues from speedboat design.

Both distance- and overlap-based similarities are useful when analyzing visual relationships between various categories in different scenarios. However, these two should work together to provide more convincing results. Being visually similar makes analogical inferences easy, and being categorically different makes the potential analogy across categories novel. One important finding is the detection of bridge categories allows both effectiveness and novelty to be obtained at the same time and may resolve the "analogical distance" dilemma as suggested by prior studies [37,72]: near-field stimuli are more effective, while far-field stimuli offer novelty. A bridge category is an analogy located in a "sweet spot" proposed by Fu et al. [37], which can offer a strategy to avoid visual fixation and find visual stimuli from long-distance analogies.

From a designer's point of view, the visual presentation of the latent space shown in Figs. 9–14 can be highly effective for the designer in choosing potential, inspiring visual cues either systematically or randomly. Upon viewing the 2D distributions of sketches like Figs. 9 and 10, a designer may intentionally choose a dataset with categories clearly from diverse taxonomies or he/she may select the one that holds closely related sketches. Making a targeted selection, i.e., clicking a colored dot on the chart or on the sketch map, allows the designer to knowingly expand her thinking toward potentially fruitful directions. Besides, visual assistance, like Fig. 10, provides the designer with a tool to explore the overlap-similarity space that has the potential to offer multilayer expansions of thinking for the designer. Furthermore, the grouping matrix displays like Figs. 12 and 14 allow designers to quickly access closely related groups of sketches which may impact designers' analogy making differently compared to single visual cue-based stimulation. Future human subject-based studies are needed to verify the effectiveness of these human augmentation strategies.

7 Conclusions

In this article, a CAVAS framework is proposed, and a deep learning-based computational model CAVAS-DL is introduced as a potential human design augmentation tool to assist human visual analogy making. The CAVAS framework extends the GSP creative stimulation model into the human–computer interaction context, and the CAVAS-DL model has demonstrated the potential of sketch-and-image–based visual analogy support.

The CAVAS-DL model is composed of a CNN-based variational autoencoder coupled with deep clustering embedding. The results from the computational and experimental studies have demonstrated CAVAS-DL's excellent capabilities in learning shape feature presentations and using distance- and overlap-based similarity to analyze visual relationships of the learned presentations of sketch categories. The application of our computational tool can potentially provide strategies to designers for enhancing their visual analogy-finding capabilities. In summary, the main contributions of this paper are as follows:

- A CAVAS support framework is introduced that extends our previous human-thinking GSP creative simulation model [1] into a human–computer interactive thinking framework. The key functional components and processes are identified for augmenting designers' visual analogical thinking processes.
- An unsupervised deep learning methodology is introduced that combines a CNN-based shape feature extraction algorithm with a deep embedded clustering model that achieves the best feature capturing and clustering simultaneously.
- Distance- and overlap-based similarities are introduced and applied to analyze visual relationships between categories.

Short- and long-distance analogies can be identified based on visual similarity. The detection of bridge categories provides a way to find long-distance analogies, which can be applied to explore testable visual cues for human-based studies of the visual analogy-making process.

In addition, the visualization of the latent space of sketches provides testable tools for human-based behavioral studies. Extensive experiments have been conducted that demonstrate the effectiveness and robustness of our proposed computational tool in exploring desired visual cues based on various given categories, which has made it possible for us to conduct human-based behavioral studies on visual analogy making.

A drawback of the CAVAS-DL model is the need to balance the weight ratio of reconstruction loss and clustering loss. It means one needs to determine the weight of clustering loss in Eq. (5). Searching for an appropriate value for the weight can take time since the model needs to be trained many times. Besides, the threshold ε to determine short- and long-distance analogies and the threshold φ to determine bridge categories are set based on the distance- and overlap-based similarity matrices and domain knowledge. More work needs to be done to efficiently find optimal values for them. Our ongoing work includes the investigation of how to determine the optimal cluster number for the clustering layer and how to use the learned sematic or functional meaning behind shapes to support visual analogy. One outstanding issue to be addressed is to evaluate how effective the visual cues explored by CAVAS-DL can be in stimulating designers' visual analogy making for generating more and better ideas. The literature review has led us to the conviction that the shape pattern is the key to bringing ideas in similar and different categories into the analogy-making context. The resulting tool Cavas-DL of this research has made it possible for us to move to the next step of conducting human subject-based design experiments to evaluate the effectiveness of computational support for visual analogy making in design.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

References

[1] Jin, Y., and Benami, O., 2010, "Creative Patterns and Stimulation in Conceptual Design," Artif. Intell. Eng. Des. Anal. Manuf, **24**(2), pp. 191–209.

[2] Goldschmidt, G., 2001, "Visual Analogy—A Strategy for Design Reasoning andLearning," *Design Knowing and Learning: Cognition in Design Education*, C. M. Eastman, W. M. McCracken , and W. C. Newstetter, eds., Elsevier, New York, pp. 199–219.

[3] Mougenot, C., Bouchard, C., Aoussat, A., and Westerman, S., 2008, "Inspiration, Images and Design: An Investigation of Designers' Information Gathering Strategies," J. Des. Res., **7**(4), pp. 331–351.

[4] Bouchard, C., Omhover, J.-f., Mougenot, C., Aoussat, A., and Westerman, S. J., 2008, "TRENDS: a Content-Based Information Retrieval System for Designers," *Proc. Design Computing and Cognition '08*, Vol. 4, Springer Science + Business Media B.V., Atlanta, GA, pp. 593–611.

[5] Chakrabarti, A., Siddharth, L., Dinakar, M., Panda, M., Palegar, N., and Keshwani, S., 2017, "Idea Inspire 3.0—A Tool for Analogical Design," Proceedings International Conference on Research Into Design, Singapore, January, Springer, pp. 475–485.

[6] Han, J., Shi, F., Chen, L., and Childs, P. R., 2018, "A Computational Tool for Creative Idea Generation Based on Analogical Reasoning and Ontology," Artif. Intell. Eng. Des. Anal. Manuf, **32**(4), pp. 462–477.

[7] Ullman, D. G., Wood, S., and Craig, D., 1990, "The Importance of Drawing in the Mechanical Design Process," Comput. Graph., **14**(2), pp. 263–274.

[8] Yang, M. C., 2009, "Observations on Concept Generation and Sketching in Engineering Design," Res. Eng. Des., **20**(1), pp. 1–11.

[9] Kokotovich, V., and Purcell, T., 2000, "Mental Synthesis and Creativity in Design: An Experimental Examination," Des. Stud., **21**(5), pp. 437–449.

[10] Goldschmidt, G., and Smolkov, M., 2006, "Variances in the Impact of Visual Stimuli on Design Problem Solving Performance," Des. Stud., **27**(5), pp. 549–569.

[11] Goldschmidt, G., 2003, "The Backtalk of Self-Generated Sketches," Des. Issues, **19**(1), pp. 72–88.

[12] Casakin, H., and Goldschmidt, G., 1999, "Expertise and the Use of Visual Analogy: Implications for Design Education," Des. Stud., **20**(2), pp. 153–175.

[13] Marshall, K. S., Crawford, R., and Jensen, D., 2016, "Analogy Seeded Mind-Maps: A Comparison of Verbal and Pictorial Representation of Analogies in the Concept Generation Process," Proceedings ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Charlotte, NC, Aug. 21–24, p. V007T006A010.

[14] Malaga, R. A., 2000, "The Effect of Stimulus Modes and Associative Distance in Individual Creativity Support Systems," Decis. Support Syst., **29**(2), pp. 125–141.

[15] Gero, J., and Yan, M., 1994, "Shape Emergence by Symbolic Reasoning," Environ. Plann. B Plann. Des., **21**(2), pp. 191–212.

[16] Oxman, R., 2002, "The Thinking eye: Visual Re-Cognition in Design Emergence," Des. Stud., **23**(2), pp. 135–164.

[17] Arnheim, R., 1997, *Visual Thinking*, University of California Press, Berkeley, CA.

[18] Chen, W., Fuge, M., and Chazan, J., 2017, "Design Manifolds Capture the Intrinsic Complexity and Dimension of Design Spaces," ASME J. Mech. Des., **139**(5), p. 051102.

[19] Chen, W., and Fuge, M., 2017, "Beyond the Known: Detecting Novel Feasible Domains Over an Unbounded Design Space," ASME J. Mech. Des., **139**(11), pp. 111405–111410.

[20] Kwon, E., Pehlken, A., Thoben, K.-D., Bazylak, A., and Shu, L. H., 2019, "Visual Similarity to Aid Alternative-Use Concept Generation for Retired Wind-Turbine Blades," ASME J. Mech. Des., **141**(3), p. 031106.

[21] Casakin, H., 2010, "Visual Analogy, Visual Displays, and the Nature of Design Problems: The Effect of Expertise," Environ. Plann. B Plann. Des., **37**(1), pp. 170–188.

[22] Herring, S. R., Chang, C.-C., Krantzler, J., and Bailey, B. P., 2009, "Getting Inspired!: Understanding How and Why Examples Are Used in Creative Design Practice," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Boston, MA, Apr. 4–9, ACM, pp. 87–96.

[23] Zhang, Z., and Jin, Y., 2020, "An Unsupervised Deep Learning Model to Discover Visual Similarity Between Sketches for Visual Analogy Support," Proceedings ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Virtual, August, p. V008T008A003.

[24] Linsey, J., Wood, K. L., and Markman, A. B., 2008, "Modality and Representation in Analogy," Artif. Intell. Eng. Des. Anal. Manuf, **22**(2), pp. 85–100.

[25] Vincent, J. F., and Mann, D. L., 2002, "Systematic Technology Transfer From Biology to Engineering," Philos. Trans. R. Soc. London, Ser. A, **360**(1791), pp. 159–173.

[26] Chakrabarti, A., Sarkar, P., Leelavathamma, B., and Nataraju, B., 2005, "A Functional Representation for Aiding Biomimetic and Artificial Inspiration of New Ideas," Artif. Intell. Eng. Des. Anal. Manuf, **19**(2), pp. 113–132.

[27] Sarkar, P., and Chakrabarti, A., 2008, "The Effect of Representation of Triggers on Design Outcomes," Artif. Intell. Eng. Des. Anal. Manuf, **22**(2), pp. 101–116.

[28] Cheong, H., Chiu, I., Shu, L., Stone, R. B., and McAdams, D. A., 2011, "Biologically Meaningful Keywords for Functional Terms of the Functional Basis," ASME J. Mech. Des., **133**(2), p. 021007.

[29] Shu, L., 2010, "A Natural-Language Approach to Biomimetic Design," AI EDAM: Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, **24**, pp. 507–519.

[30] Vattam, S., Wiltgen, B., Helms, M., Goel, A. K., and Yen, J., 2011, "DANE: Fostering Creativity in and Through Biologically Inspired Design," *Design Creativity 2010*, Taura T. and Nagai Y. ed., Springer, London, pp. 115–122.

[31] Goel, A. K., Vattam, S., Wiltgen, B., and Helms, M., 2012, "Cognitive, Collaborative, Conceptual and Creative—Four Characteristics of the Next Generation of Knowledge-Based CAD Systems: A Study in Biologically Inspired Design," Comput. Aided Des., **44**(10), pp. 879–900.

[32] Nagel, J. K., and Stone, R. B., 2012, "A Computational Approach to Biologically Inspired Design," Artif. Intell. Eng. Des. Anal. Manuf, **26**(2), pp. 161–176.

[33] Deldin, J.-M., and Schuknecht, M., 2014, "The AskNature Database: Enabling Solutions in Biomimetic Design," *Biologically Inspired Design*, A. Goel, D. McAdams, and R. Stone , eds., Springer, London, pp. 17–27.

[34] Murphy, J., Fu, K., Otto, K., Yang, M., Jensen, D., and Wood, K., 2014, "Function Based Design-by-Analogy: A Functional Vector Approach to Analogical Search," ASME J. Mech. Des., **136**(10), p. 101102.

[35] Fu, K., Cagan, J., Kotovsky, K., and Wood, K., 2013, "Discovering Structure in Design Databases Through Functional and Surface Based Mapping," ASME J. Mech. Des., **135**(3), p. 031006.

[36] Song, B., and Luo, J., 2017, "Mining Patent Precedents for Data-Driven Design: The Case of Spherical Rolling Robots," ASME J. Mech. Des., **139**(11), p. 111420.

[37] Fu, K., Chan, J., Cagan, J., Kotovsky, K., Schunn, C., and Wood, K., 2013, "The Meaning of "Near" and "far": the Impact of Structuring Design Databases and the Effect of Distance of Analogy on Design Output," ASME J. Mech. Des., **135**(2), p. 021007.

[38] Luo, J., Sarica, S., and Wood, K. L., 2021, "Guiding Data-Driven Design Ideation by Knowledge Distance," Knowl.-Based Syst., **218**, p. 106873.
[39] Sarica, S., Luo, J., and Wood, K. L., 2020, "TechNet: Technology Semantic Network Based on Patent Data," Expert Syst. Appl., **142**, p. 112995.
[40] He, Y., Camburn, B., Liu, H., Luo, J., Yang, M., and Wood, K., 2019, "Mining and Representing the Concept Space of Existing Ideas for Directed Ideation," ASME J. Mech. Des., **141**(12), p. 121101.
[41] He, Y., Camburn, B., Luo, J., Yang, M. C., and Wood, K. L., 2019, "Visual Sensemaking of Massive Crowdsourced Data for Design Ideation," Proceedings of the Design Society: International Conference on Engineering Design, Delft, The Netherlands, Aug. 5–8, Cambridge University Press, pp. 409–418.
[42] Gonçalves, M., Cardoso, C., and Badke-Schaub, P., 2014, "What Inspires Designers? Preferences on Inspirational Approaches During Idea Generation," Des. Stud., **35**(1), pp. 29–53.
[43] Macomber, B., and Yang, M., 2011, "The Role of Sketch Finish and Style in User Responses to Early Stage Design Concepts," Proceedings ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Washington, DC, January, pp. 567–576.
[44] Linsey, J. S., Clauss, E., Kurtoglu, T., Murphy, J., Wood, K., and Markman, A., 2011, "An Experimental Study of Group Idea Generation Techniques: Understanding the Roles of Idea Representation and Viewing Methods," ASME J. Mech. Des., **133**(3), p. 031008.
[45] Atilola, O., Tomko, M., and Linsey, J. S., 2016, "The Effects of Representation on Idea Generation and Design Fixation: A Study Comparing Sketches and Function Trees," Des. Stud., **42**, pp. 110–136.
[46] McKoy, F. L., Vargas-Hernández, N., Summers, J. D., and Shah, J. J., 2001, "Influence of Design Representation on Effectiveness of Idea Generation," Proceedings of ASME DETC, Pittsburgh, PA, September, pp. 9–12.
[47] Casakin, H., 2004, "Visual Analogy as a Cognitive Strategy in the Design Process: Expert Versus Novice Performance," J. Des. Res., **4**(2), p. 124.
[48] Casakin, H. P., and Goldschmidt, G., 2000, "Reasoning by Visual Analogy in Design Problem-Solving: The Role of Guidance," Environ. Plann. B Plann. Des., **27**(1), pp. 105–119.
[49] Stiny, G., 1993, "Emergence and continuity in shape grammars,", Proceedings CAAD Futures, U. Flemming, and S. Van Wyk, eds., North-Holland, Amsterdam, pp. 37–54.
[50] Zijian, Z, and Yan, J, 2021, "Toward Computer Aided Visual Analogy Support (CAVAS): Augment Designers Through Deep Learning," Proceedings ASME 2021 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Virtual, August.
[51] Zijian, Z, and Yan, J, 2022, "Data-Enabled Sketch Search and Retrieval for Visual Design Stimuli Generation," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, **36**, p. e25.
[52] Bell, S., and Bala, K., 2015, "Learning Visual Similarity for Product Design With Convolutional Neural Networks," ACM Trans. Graph., **34**(4), pp. 1–10.
[53] Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., and Carin, L., 2016, "Variational Autoencoder for Deep Learning of Images, Labels and Captions," Proceedings Advances in Neural Information Processing Systems, Barcelona, Spain, December, pp. 2352–2360.
[54] Yu, Q., Yang, Y., Song, Y.-Z., Xiang, T., and Hospedales, T., 2017, "Sketch-a-Net That Beats Humans," Int. J. Comput. Vis., **122**, pp. 411–425.
[55] Karimi, P., Maher, M. L., Davis, N., and Grace, K., 2019, "Deep Learning in a Computational Model for Conceptual Shifts in a Co-Creative Design System," arXiv preprint arXiv:1906.10188.
[56] Jiang, S., Luo, J., Ruiz-Pava, G., Hu, J., and Magee, C. L., 2021, "Deriving Design Feature Vectors for Patent Images Using Convolutional Neural Networks," ASME J. Mech. Des., **143**(6), p. 061405.
[57] Jiang, S., Luo, J., Pava, G. R., Hu, J., and Magee, C. L., 2020, "A CNN-Based Patent Image Retrieval Method for Design Ideation,"
[58] Ha, D., and Eck, D., 2017, "A Neural Representation of Sketch Drawings," arXiv preprint arXiv:1704.03477.
[59] Kingma, D. P., and Welling, M., 2013, "Auto-Encoding Variational Bayes," arXiv preprint arXiv:1312.6114.
[60] Schuster, M., and Paliwal, K. K., 1997, "Bidirectional Recurrent Neural Networks," IEEE Trans. Signal Process., **45**(11), pp. 2673–2681.
[61] Reed, S., van den Oord, A., Kalchbrenner, N., Colmenarejo, S. G., Wang, Z., Chen, Y., Belov, D., and de Freitas, N., 2017, "Parallel Multiscale Autoregressive Density Estimation," Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, July, Vol. 70, JMLR, pp. 2912–2921.
[62] Chen, Y., Tu, S., Yi, Y., and Xu, L., 2017, Sketch-pix2seq: A Model to Generate Sketches of Multiple Categories," arXiv preprint arXiv:1709.04121.
[63] Goldschmidt, G., 1994, "On Visual Design Thinking: The vis Kids of Architecture," Des. Stud., **15**(2), pp. 158–174.
[64] Jongejan, J., Rowley, H., Kawashima, T., Kim, J., and Fox-Gieg, N., 2016, Mount View, CA, https://quickdraw.withgoogle.com/, Accessed February 17, 2018.
[65] Xie, J., Girshick, R., and Farhadi, A., 2016, "Unsupervised Deep Embedding for Clustering Analysis," Proceedings of the International Conference on Machine Learning, New York, June, pp. 478–487.
[66] Van der Maaten, L., and Hinton, G., 2008, "Visualizing Data Using t-SNE," J. Mach. Learn. Res., **9**(11), pp. 2579–2605.
[67] Nigam, K., and Ghani, R., 2000, "Analyzing the Effectiveness and Applicability of co-Training," Proceedings of the Ninth International Conference on Information and Knowledge Management, McLean, VA, November, pp. 86–93.
[68] Kuhn, H. W., 1955, "The Hungarian Method for the Assignment Problem," Nav. Res. Logist. Q., **2**(1–2), pp. 83–97.
[69] Vetter, P., and Newen, A., 2014, "Varieties of Cognitive Penetration in Visual Perception," Conscious. Cogn., **27**, pp. 62–75.
[70] Meyer, B. H., Pozo, A. T. R., and Zola, W. M. N., 2022, "Global and Local Structure Preserving GPU t-SNE Methods for Large-Scale Applications," Expert Syst. Appl., **201**, p. 116918.
[71] Agis, D., and Pozo, F., 2019, "A Frequency-Based Approach for the Detection and Classification of Structural Changes Using t-SNE," Sensors, **19**(23), p. 5097.
[72] Srinivasan, V., Song, B., Luo, J., Subburaj, K., Elara, M. R., Blessing, L., and Wood, K., 2018, "Does Analogical Distance Affect Performance of Ideation?," ASME J. Mech. Des., **140**(7), p. 071101.