SpaceSheet: Navigating Conceptual Space with a Spreadsheet Interface

Tom White and Bryan Loh

School of Design
Victoria University of Wellington
Wellington, New Zealand
{tom.white@vuw.ac.nz, lohbrya@myvuw.ac.nz}

Abstract

We introduce a new spreadsheet based interface called SpaceSheets for creating novel images and other media. Unlike traditional digital tools, ours is parameterized entirely by a neural network with no preprogrammed rules or knowledge representations. The capability of SpaceSheets to support visual exploration and communication is demonstrated within the context of several domains including facial images, fonts, and english words. SpaceSheets is demonstrated to support the experimentation and exploration of latent spaces enabling more effective design experimentation.

Introduction

Problem solving can be viewed as a search for a solution within a space. In design, this process involves generating solutions and evaluating their consequences relative to goals and constraints (Simon 1995). These experiments are enabled through representations in the form of drawings and diagrams. Computational design tools enable users to construct and manipulate representations digitally. These tools often impose a high cost to design experimentation due to the mismatch between low-level design operations in expressing more abstract design intent.

Generative models learn more compact representations of the training data in a vector space of latent variables. Latent variables are sampled from high-dimensional latent space and can be decoded back into observable values. Additionally, semantic operations can be performed within latent space using vector arithmetic (White 2016).

Spreadsheet interfaces are a ubiquitous part of office productivity suites. They enable users to perform experimental calculations using a set of formulae which define relationships spatially. Automatic recalculation supports experimentation by enabling users to observe the results of their actions immediately and act accordingly.

We developed SpaceSheet (Figure 1) to leverage the familiarity and power of spreadsheet interfaces for the purpose of design experimentation within latent space. It has been adapted to enable non-experts to explore and experiment within latent spaces.
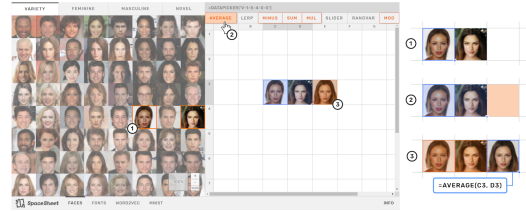

Figure 1: The SpaceSheet being used to perform an average between two latent variables

Background

Conceptual Spaces

Generative models are a popular approach to unsupervised machine learning. Generative neural network models are trained to produce data samples that resemble the training set (Karpathy et al. 2016). Because the number of model parameters is significantly smaller than the training data, the models are forced to discover efficient data representations. These models are sampled from a set of latent variables in a high-dimensional space, called a latent space. Latent space can be sampled to generate observable data values. Learned latent representations often also allow semantic operations with vector space arithmetic (Figure 2), a phenomenon discovered previously in the latent space of language models (Mikolov et al. 2013).

Generative models are often applied to datasets of images. Two popular generative models for image data are the Variational Autoencoder (Kingma and Welling 2013) (VAE) and the Generative Adversarial Network (Goodfellow et al. 2014) (GAN). VAEs use the framework of probabilistic graphical models with an objective of maximizing a lower bound on the likelihood of the data. GANs instead formalize the training process as a competition between a generative network and a separate discriminative network. Though these two frameworks are very different, both construct high-dimensional latent spaces that can be sampled to generate images resembling training set data. Moreover, these latent spaces are generally highly structured and can enable complex operations on the generated images by simple vector space arithmetic in the latent space (Larsen,

Figure 4: Decoupling attribute vectors for smiling (x-axis) and mouth open (y-axis) allows for more flexible latent space transformations. Input shown at left with reconstruction adjacent. Using a VAE model from (Lamb, Dumoulin, and Courville 2016)
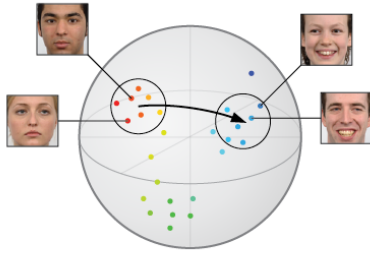
Figure 2: Schematic of the latent space of a generative model. In the general case, a generative model includes an encoder to map from the feature space (here images of faces) into a high-dimensional latent space. Vector space arithmetic can be used in the latent space to perform semantic operations. The model also includes a decoder to map from the latent space back into the feature space, where the semantic operations can be observed. If the latent space transformation is the identity function we refer to the encoding and decoding as a reconstruction of the input through the model.

Sønderby, and Winther 2015).

In the latent space of generative models, many high-level attributes can be represented as a vector (Figure 3). Using techniques from (White 2016), multiple attributes can be decoupled further to create a visualization of possible states across multiple semantic vectors (Figure 4). For example, when trained on a dataset of portraits, latent vectors can be computed for "smiling" and "mouth open" which then applied to new face images.

Figure 3: Traversals along the smile vector using a GAN model from (Dumoulin et al. 2016)

Prior to the discovery of neural network latent spaces supporting semantic operations, cognitive science had hypothesized the existence of knowledge representations that were primarily geometric instead of symbolic. One primary proponent was Gärdenfors who proposed a framework of "Conceptual Spaces" as structured multi-dimensional feature spaces to support modeling information processes such as concept learning and prototype theory (Gärdenfors 2011). Notably, conceptual spaces were proposed as a model of how people structure concepts, independent of any proposed computational implementation of how they might come about.

We adapt the terminology and claim that latent spaces of generative neural networks function as conceptual spaces
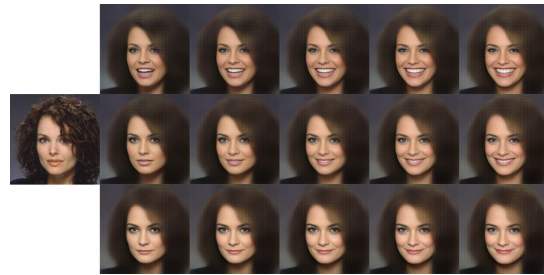
which can be used as non-symbolic knowledge representation layers in other tools. With this framework, we examine the ability of this representation layer built from the latent space of a generative neural network model to support a new type of spreadsheet interface tool. The tool itself is domain independent and is shown to be useful in several domains. In exploring these particular domains, our tool constructs subspaces of the larger conceptual space of possibilities as a parameter space of a spreadsheet driven exploration tool.

Supporting Design Experimentation

Design principles have been identified by (Resnick et al. 2005) and (Terry and Mynatt 2002) for user interfaces to support design experimentation and exploration.

These principles can be summarised by the three user interface requirements proposed in Design Principles for Tools to Support Creative Thinking (Resnick et al. 2005) (paraphrased): It must be very easy to try things out and then backtrack when unsuccessful. Tools should be 'self-revealing' in what they can achieve. Make it very fast to sketch out different alternatives

These principles are supported by (Terry and Mynatt 2002) where they identify three activities in the process of reflection-in-action (Schon 1984) that should be supported by user interfaces for design experimentation. They are: Near-Term Experimentation, Generating Variations, and Evaluation.

Near-Term Experimentation is used to describe actions which intend to "discover and instantiate the next move" (Terry and Mynatt 2002, p. 39). In a user interface, users would make hypotheses about the next action to be made, and test their hypothesis by "invoking a command and adjusting its settings to achieve the imagined effect". The users would then "either accept the command, tweak the parameters more, or undo it and try another tact" (Terry and Mynatt 2002, p. 40).

Variations are created by the designer to explore alternatives deeply. It enables them "to better understand the problem, its boundaries, and potential solutions" (Terry and Mynatt 2002, p. 40). An example of this is where design-

ers make "multiple variations of a specific component by creating them side-by-side on a large canvas ... and iterate on promising versions to arrive at an acceptable solution" (Terry and Mynatt 2002, p. 40).

Users need to evaluate their progress as they work on a task. This happens after near-term experiments, as well as after generating variations: "the moment in which the individual reassesses the problem and their understanding of it, before making the next move" (Terry and Mynatt 2002, p. 40).

Spreadsheet as a Design tool

Spreadsheets may seem like an unlikely design tool. However, the ability to express relationships between cells make it functionally suited to express operations in latent space. Additionally, it satisfies the three user requirements for software to support design experimentation — Near-Term Experimentation, Generating Variations, and Evaluation as proposed by (Terry and Mynatt 2002).

Near-term experimentation is supported by the automatic updating feature of the spreadsheet. Users are able to set up scenarios of logic and calculate the results to 'what if' questions instantly by modifying the cell values. This establishes a tight feedback loop between the user's actions and its implications. When coupled with the ability to undo actions, it enables users to discover and instantiate moves, and backtrack if the results are unsatisfactory.

The generation of variations is supported by enabling users to duplicate instances of data onto other cells within the document. These copies can then be modified independently from the original data.

Evaluation is supported by enabling users flexibility in how they choose to organise data in the document. Users can set up custom templates in a layout which best supports their preferences and the problem to be solved.

In addition to their promise in supporting design experimentation, spreadsheet software is well-established within office productivity suites. Users with an understanding of how conventional spreadsheets function are able to transfer their understanding to the use of the design tool.

SpaceSheet

SpaceSheet consists of a data picker exposing latent variables to operate with and a spreadsheet to define operations between the variables. In both, latent variables are decoded into observable images.

Data Picker

The data picker is a predetermined set of latent variables which have been organized into a grid. The set of variables in the data picker act as the points of reference from which the latent space can be explored from. Diversity has been prioritized in the selected set to maximize the variety of possible outcomes that can be explored. Multiple data pickers have also been implemented as tabs to provide various pre-baked distributions of latent variables.

Spreadsheet

The spreadsheet is the main workspace of the tool. It enables users to express relationships between cells using formulae. Operations between cells containing latent variables are computed with vector arithmetic, and its result is decoded into an image. Common operations can be defined by clicking on buttons at the top of the spreadsheet. These buttons are selection-aware, and highlight to suggest operations based on the selected cells. A live SpaceSheets demo is available online[1] and the appendix contains a list of supported operations and sample workflows.

Applications

Figure 5: SpaceSheet with Font Model

Initial efforts are focused on experimenting in various domains to encourage the development of a general-purpose model agnostic set of operations. A SpaceSheet to explore a generative model of fonts (Bernhardsson 2015) has been implemented to be used as a design tool (Figure 5). User testing indicated that the tool enabled designers and non-designers alike to explore design variations easily (Loh 2018).

Figure 6: SpaceSheet with word2vec

---

[1]https://vusd.github.io/spacesheet/

The concepts have also been extended to domains other than images and with models that are not generative, such as the Word2Vec model (Mikolov et al. 2013). This version of the SpaceSheet can be used to find word analogies and perform interpolations using nearest neighbors (Figure 6).

A SpaceSheet has been created to enable the exploration of the BigGAN model (Brock, Donahue, and Simonyan 2018). In this implementation, the primary DataPicker for this implementation has been curated to enable users to experiment with a variety of image classes (Figure 7).
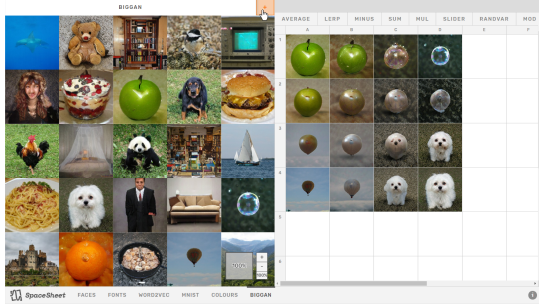

Figure 7: BigGAN SpaceSheet with a generic DataPicker across all image classes

Custom DataPickers of other classes, or combinations of other classes can be created using the DataPicker creator (Figure 8). The DataPicker creator enables users to a) explore and select one or more classes from a searchable, hierarchically organised tree checklist, b) control the amounts of each class to composite in the resulting class, and c) preview example reconstructions of the resulting class before creating a DataPicker of the resulting class. Once created, this new custom DataPicker will be available for use in the spreadsheet (Figure 9).
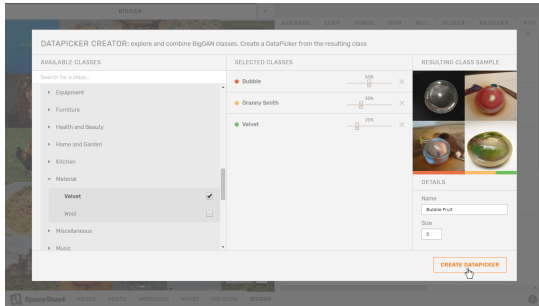

Figure 8: BigGAN SpaceSheet custom DataPicker interface

Evaluation

User testing of SpaceSheets on a model of fonts (Loh 2018) revealed that the tool enabled a novel method to experiment with designs. Users explore design possibilities from a top-down approach by deriving meaning and navigating within a
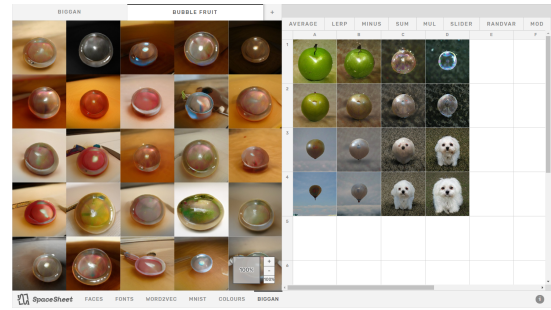

Figure 9: BigGAN SpaceSheet with a custom DataPicker made from combining a user-provided ratio of the "Bubble", "Granny Smith", and "Velvet" image classes

preconstructed model, rather than constructing a model from the bottom-up.

This method of working was reported to be more supportive of design exploration, more efficient, and capable of enabling non-designers to explore design possibilities. Unsurprisingly, it required new skills and intuition to be used to its full effect. A lack of knowledge in deriving and applying attribute vectors from latent space limited users' expressivity and control over their experiments. Due to this, interpolation was found to be the most intuitive and common method to arrive at search targets.

Expressing low-level transformations such as positioning and scale through SpaceSheet often resulted in distorted reconstructions which did not match the expectations of the user. This is attributed to a mismatch in the high-level probabilism of sampling latent spaces is an ill-fit to express concrete design intent. However, this uncertainty has been reported to be serendipitous when distortions in the reconstruction added to the aesthetics of the design.

Discussion

SpaceSheets explores the potential of latent spaces to be used as a tool for design experimentation. The research finds it to enable a novel method to work with designs which supports more efficient, high-level design experimentation to designers and non-designers alike.

User intuition and skill in deriving meaning from latent spaces is fundamental to conduct design experiments with a fine level of control. This intuition can be considered a skill which can be developed through continued experience with the flexible, low-level interface provided by the SpaceSheet. Although latent spaces enable designers to express more meaningful design operations computationally, it provides redundant uncertainty for low-level design operations. It is with this understanding that latent spaces are best considered as a complementary new primitive to build smarter design tools.

References

Bernhardsson, E. 2015. deep-fonts: Generate fonts using deep learning.

Brock, A.; Donahue, J.; and Simonyan, K. 2018. Large scale GAN training for high fidelity natural image synthesis. *CoRR* abs/1809.11096.

Dumoulin, V.; Belghazi, I.; Poole, B.; Mastropietro, O.; Lamb, A.; Arjovsky, M.; and Courville, A. 2016. Adversarially Learned Inference. *ArXiv e-prints*.

Gärdenfors, P. 2011. Semantics based on conceptual spaces. In *Logic and Its Applications - 4th Indian Conference, ICLA 2011, Delhi, India, January 5-11, 2011. Proceedings*, 1–11.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A. C.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2672–2680.

Karpathy, A.; Abbeel, P.; Brockman, G.; Chen, P.; Cheung, V.; Duan, R.; Goodfellow, I.; Kingma, D.; Ho, J.; Houthooft, R.; Salimans, T.; Schulman, J.; Sutskever, I.; and Zaremba., W. 2016. Generative models. Technical Report 1, OpenAI.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *CoRR* abs/1312.6114.

Lamb, A.; Dumoulin, V.; and Courville, A. 2016. Discriminative Regularization for Generative Models. *ArXiv e-prints*.

Larsen, A. B. L.; Sønderby, S. K.; and Winther, O. 2015. Autoencoding beyond pixels using a learned similarity metric. *CoRR* abs/1512.09300.

Loh, B. 2018. Spacesheets: Design experimentation in latent space.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Resnick, M.; Myers, B.; Nakakoji, K.; Shneiderman, B.; Pausch, R.; Selker, T.; and Eisenberg, M. 2005. Design principles for tools to support creative thinking.

Schon, D. 1984. The reflective practitioner: How professionals think in action.

Simon, H. 1995. Problem Forming, Problem Finding, and Problem Solving in Design. 245–257.

Terry, M., and Mynatt, E. 2002. Recognizing creative needs in user interface design. 38—44.

White, T. 2016. Sampling generative networks. *CoRR* abs/1609.04468.

Appendix: Implementation Details
Supported Operations

| Operation	Description	Formula
Sum	Adds a list of numbers / variables	`SUM(val1, val2, val3, ...)`
Minus	Subtracts two numbers / variables in sequence	`MINUS(val1, val2)`
Multiply	Multiplies a list of numbers / variables	`MUL(val1, val2, val3, ...)`
Linear Interpolation	Calculates the value in between two numbers / vectors at a specified amount	`LERP(from, to, amount)`
Average	Calculates the average of a list of numbers / vectors	`AVERAGE(val1, val2, val3, ...)`
Distance	Calculates the euclidean distance between two numbers / vectors	`DIST(val1, val2)`
Modulate	Creates a scrubbing interface which can modulate a cell	`MOD(cell, degree, radius)`
Random Variable	Creates a random latent variable	`RANDVAR(seed)`
Slider	Creates a number which is controlled by a slider element	`SLIDER(min, max[, step])`

Interactive Cell Types
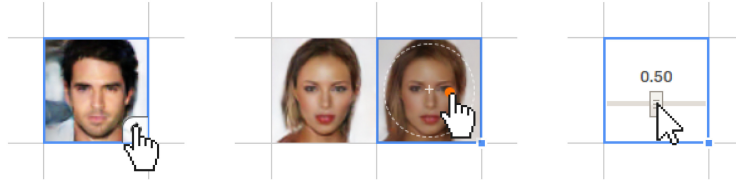
Figure 10: RANDVAR, MOD and SLIDER cells.

Several alternative cell types have been implemented to create interface elements which support more effective search and exploration. These are instantiated by the operations:

`RANDVAR(seed)`

The RANDVAR (random variable) cell instantiates a latent variable from a random seed. This enables users to operate using latent variables beyond the limited set afforded by the Data Picker. A button displays when the cell is hovered over which enables users to randomise the cell directly.

`MOD(base, degree, distance)`

The MOD (modulate) cell exposes a joystick interface which enables users to scrub locally around a given latent variable to arrive at similar latent variables. The degree of difference can be controlled by the joystick's distance from the center of the cell.

`SLIDER(min, max [,step])`

The slider cell enables users to create a number controlled by a slider element.

Example Workflows

Interpolation

Figure 11: An interpolation between two latent variables

Extrapolation

Figure 12: Extrapolating from two points.

Extrapolating from latent variables can be used to emphasise attributes which vary between its anchors. In this example, the difference between the highlighted anchors - blond hair, large smile, etc. - have been emphasised by extrapolating beyond the end anchor.
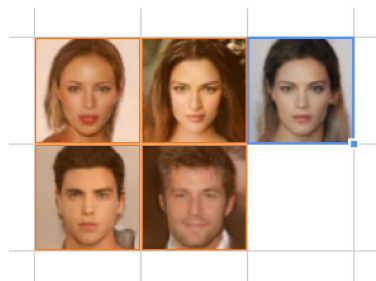
Averaging

Figure 13: Calculating the average reconstruction of a group of latent variables
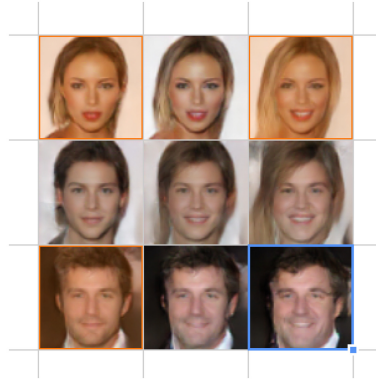
Analogy

Figure 14: An analogical construction. The bottom right cell applies the difference between the top cells to the cell on the bottom-left

Given three reconstructions (top-left, top-right, bottom-left), the SpaceSheet calculates the bottom-right corner by analogy. This is achieved by applying the difference between the top variables to the bottom-left variable. In this example, a toothy grin has been applied to the man.
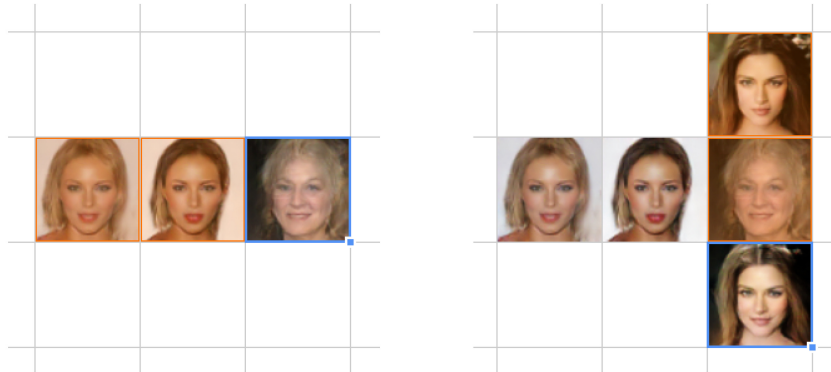
Attribute Vectors

Figure 15: Isolating a 'blonde' vector by subtraction (left). Adding the attribute vector to a new latent variable (right)

Specific attributes can be applied as operations to latent variables. Attribute vectors can be isolated by subtracting a latent vector with desired attributes with one without the attributes. This attribute vector can be added to another latent variable to apply the isolated attribute. The example image shows this two-step process. In the first, a 'blonde' attribute vector has been isolated by computing the difference between the highlighted cells. This vector is then applied in the right image by addition. The result is a more blonde version of the initial latent variable.