Université de Montréal

Emergent structure in representation learning – Generalization & Application

par

Samuel Lavoie

Département d'information et de recherche opérationnelle

Faculté des arts et des sciences

Proposition de sujet de thèse en apprentissage machine

February 25, 2022

Université de Montréal

Faculté des arts et des sciences

Cette proposition de sujet de thèse intitulée

Emergent structure in representation learning – Generalization & Application

présentée par

Samuel Lavoie

a été évaluée par un jury composé des personnes suivantes :

Nom du président du jury

(président-rapporteur)

Aaron Courville

(directeur de recherche)

Nom du membre de jury

(membre du jury)

(examinateur externe)

(représentant du doyen de la FESP)

Résumé

Dans ce travail, nous étudions comment apprendre des représentations des données qui ont certaines structures dans le but d'améliorer la généralisation, ainsi que de nous intéresser à l'utilisation de ces représentations structurées dans le cadre d'applications.

Dans le premier projet présenté dans ce document, nous proposons une méthode pour apprendre une représentation qui capture la sémantique partagée par les données provenant de différents domaines. Nous utilisons alors cette représentation dans le contexte de transfert de domaines des images sans supervision dans le but d'informer le modèle de la sémantique étant transférée ainsi que de contraindre l'apprentissage à préserver cette sémantique. Cette méthode permet un transfert d'attributs de sémantique à plus haut niveau comme les catégories d'objets.

Dans le second projet, nous proposons un module que nous plaçons dans les modèles d'apprentissage à auto-supervision dans le but d'induire une discrétisation douce. Nous démontrons que ce module améliore la généralisation systématique ainsi que la généralisation à des domaines différents que celui d'entraînement.

Mot clés: Apprentissage de representation non supervisé, généralisation.

Abstract

In this work, we are concerned with learning representation with specific structures to improve generalization and how to leverage these representations in the context of applications.

In the first project, we propose a method to learn a representation that captures the shared semantics of samples across different domains. We then leverage this representation in the context of Unsupervised Domain Translation to inform the model on the semantics being transferred and to constrain the learning procedure to preserve said semantics. This method allows translation between higher-order semantic attributes such as object categories.

The second project proposes a drop-in module to induce soft-discretization in self-supervised learning. We demonstrate increased systematic out-of-distribution generalization and domain generalization performance.

Keywords. Representation learning, Unsupervised learning, Out-of-distribution generalization.

Contents

List of tables

List of figures

Remerciements

In the last years, I learned a lot about how to conduct research, write papers and collaborate. This would not have been possible without the help of my colleagues here at Mila. I want to acknowledge Faruk Ahmed, who taught me a great deal about writing and storytelling. I also want to thank Max, Christos and Ankit for the opportunity to collaborate with them on the project presented in chapter 3. It is a privilege to be learning with you guys. Finally, I want to thank Aaron Courville for guiding me throughout this journey. You are a great mentor, and I am grateful for the chance to be working with you.

Introduction

Despite deep learning showing increasingly impressive applications, there is not a consensus as to why it currently works as well as it does. Some skeptical actors claim that all deep learning does is "curve fitting." While there is truth in this claim, we will entertain the idea that this curve fitting process can lead to useful emergent properties such as structure in the organization of the parameters of the model as well as structure representing observable attributes of the sample in the hidden representation of the network.

We define the emergence of a property P from a system S with dynamics D, in the weak sense of the word, as follows [Bedau, 1997]: "A [property] P of a system S with microdynamics D is emergent if and only if P can be derived from D and S's external condition but only by simulation". We illustrate this concept with two examples:

Example 1. Conway's Game of Life. Conway's Game of Life [Gardner, 1970] is a grid-world game where each cell evolves according to the state of its neighbours. Depending on the initial conditions of the system – which are set by an external agent, and the rules [1] that defines the dynamics – different shapes and behaviours that can emerge. For example, we see that a periodic system emerges in Figure 1b.

□

[1] The production rules are defined as follows [Gardner, 1970]: 1- any live cell with two or three live neighbours survives. 2- Any dead cell with three live neighbours becomes a live cell. 3- All other live cells die in the next generation.

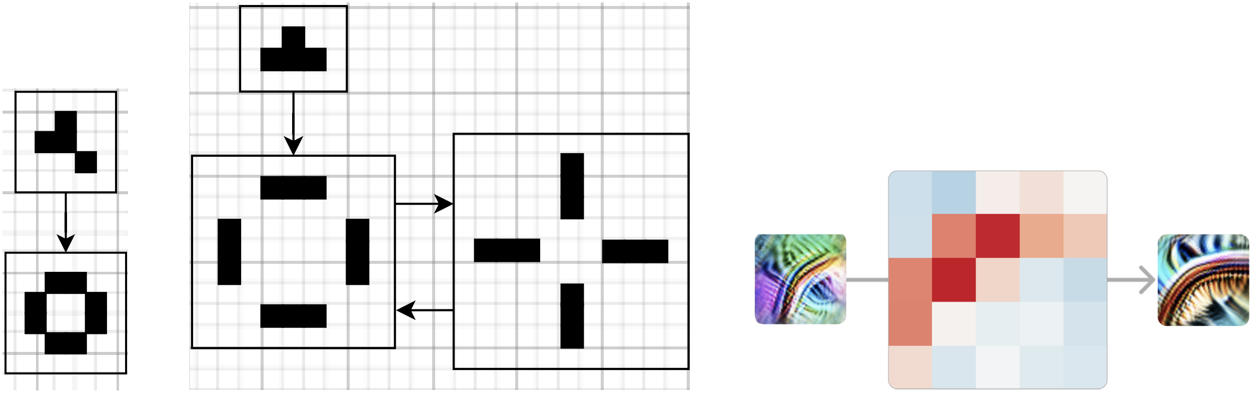


Fig. 1. (a) **Conway's game of life.** The system converge to a static loop after 4 steps (b) **Conway's game of life.** Given a slightly different set of initial, a periodic cycle emerges after 10 steps. (c) **Curve detectors [Olah et al., 2020]** An edge detector emerges in the parameters of the kernel of a convolution (red are positive weights and blue are negative weights) in the second layer of a VGG network after the training on ImageNet.

This example demonstrates that emergent properties can arise in a simple setup. The following example is related to a property that we observe in deep learning when training convolutional neural networks on vision tasks.

Example 2. Curve detector in a convolutional neural network. The training of a neural network is a system with dynamics governed by the optimization procedure. The optimization procedure is typically a greedy algorithm, notably Stochastic Gradient Descent, that aims at finding the parameters θ of a function f that minimizes a specific objective function $\mathcal{L}$. More precisely, the dynamics of the parameters of a neural network f_θ is governed by the following update rule:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(f_\theta, \mathcal{X}, \cdot).$$

The external conditions include the definition of the objective function $\mathcal{L}$, the training $\mathcal{X}$, the initialization scheme of θ, the learning rate α and other conditions such as the data augmentation and other hyperparameters that we denote with the dot $(\cdot)$.

Given the right set of conditions, we can observe different useful functions emerge. Examples include a dog detector if a convolutional neural network [LeCun et al.] [2] is trained on a dataset with images of dogs, or more generally an edge detector as depicted in Figure 1c.

□

The word "structure" will be used throughout this document qualitatively to describe *interesting* organization of data or parameters. For example, embeddings where natural images are clustered according to their label – allowing a linear classifier onto – have structure. The organization of parameters into a curve detector in a convolution kernel has structure, and the composition of such kernels to generate a dog detector also has structure. While there are efforts into bringing a more grounded notion of structure to machine learning using ideas from geometry [Bronstein et al., 2021], abstract algebra [de Haan et al., 2020] and topological data analysis [Carlsson and Gabrielsson, 2018], the formal treatment of the word structure is out of the scope of this document. Nonetheless, we still use this word since it captures the essence of the meaning that we want to convey.

This document will initially discuss current methods proposed in the deep learning literature for encouraging the emergence of structure. Notably, we will discuss different architectural and objective functions and the properties that they lead to. In the same chapter, we will discuss how the community studying artificial language emergence has approached the question of the emergence of structure in language. This discussion will only be used in chapter 3, where we will discuss our ongoing work and potential future works. In chapter 2, we will tackle the problem of preserving semantics in Unsupervised Domain Translation. We

[2] A convolutional neural network is a function defined mostly by a composition of convolutions. The composition of convolutions is often intertwined with element-wise non-linearity such as ReLU Agarap [2019]

will approach the problem by first proposing a method for learning a categorical representation of the shared semantics between two domains. This representation will be leveraged in Unsupervised Domain Translation frameworks to penalize the network for not preserving the learned semantics. Finally, in chapter 3, we will explore how to improve systematic generalization and domain generalization using ideas from the language emergence community. We will present a current work where we attempt to impose structure in self-supervised learning models by imposing a soft discretization bottleneck. Finally, we conclude with some ideas that we think are promising for further improving generalization in self-supervised learning, including the idea of using Iterated Learning and a direction for semantics identification for self-supervised learning.

Chapter 1

Structure in representation learning and artificial language

1.1. Representation learning

Representation learning is the cornerstone of deep learning. Whether we learn a model end-to-end or learn a representation that will subsequently be for a downstream application, we always aim to learn a representation that will have useful properties for our desired task. For example, take the classification task, where the objective is to minimize the error between the predicted label from a model and a label given by a human. As demonstrated by Huh et al. [2016], Yosinski et al. [2015], Zhou et al. [2015], Bau et al. [2017], Simonyan et al. [2014], in applying convolutional neural networks to computer vision, special-purpose modules tailored for detecting categories in the image emerge in the parameters of the network. Furthermore, Carbonnelle and Vleeschouwer [2020] observed that the images cluster according to the labelled categories in the hidden representations of a trained network, demonstrating emergent structure in the embedding of the samples themselves. While it is useful that the representation develops such a structure, other less desirable solutions are possible. For example, the samples could be represented by some idiosyncratic coordinates. In other words, it could memorize a dictionary of the sample and its category since this function minimizes the training error and deep networks are universal function approximators [Hornik et al., 1989]. Zhang et al. [2017] demonstrated that such as solution is possible in practice by fitting a neural network to random labels. While we would expect a classifier to classify the training samples, it would be surprising that it generalizes to samples outside of the training set. Thus we can say that this representation is not useful due to its lack of structure.

In practice, we observe that structure may arise given the correct pressures. The emergence of structural properties is governed by the microdynamics of a system and its external condition. Researchers and practitioners have been inducing structure by using clever objectives and architectures. However, it has recently been demonstrated that increasing the scale of data can lead to an improvement in generalization and few-shot learning capabilities [Brown et al., 2020].

In this document, we will study how imposing the right set of objectives can lead to semantic preserving Unsupervised Domain Translation in chapter 2. Furthermore, we will demonstrate that a drop-in architectural module that imposes a soft-discretization improves systematic generalization and robustness to domain shift in chapter 3. We will now review some objectives and architectural choices relevant to the works presented in the sequel.

1.1.1. Architecture – Convolutional Neural Networks

The main idea of a Convolutional Neural Networks (CNN) [LeCun et al.] is to use a convolution operator in place of matrix multiplication in at least one layer of a neural network. Since we are primarily concerned with computer vision application in this document, we define the convolutional operation (more precisely the cross-correlation operation) for $2D$-grids as follows:

$$s(i, j; I, K) = \sum_{m=0}^{M} \sum_{n=0}^{N} I(i + m, j + n) K(m, n), \tag{1.1.1}$$

where I is the image and K is the kernel applied on I with width M and height N. We notice that for M and N smaller than the width and the height of I respectively, the number of parameters of a convolution operator is smaller than the number of parameters for matrix multiplications. The number of parameters is constant with respect to the input size for the convolution, whereas the number of parameters scales quadratically with the input size for a fully connected layer. The convolution reduces the complexity of the network, not only because it has fewer parameters than its fully connected counterpart but also because it can be represented by a doubly block circulant matrix and is thus strictly a subset of the possible fully connected matrices, reducing the number of possible solutions. Using convolutions instead of fully connected layers improves generalization in vision tasks, demonstrating that such compression is a good inductive bias for these tasks.

Relatedly, residual connections [He et al., 2015] are often used in conjunction with convolution neural networks. Defined as follows

$$\boldsymbol{z}' := \mathcal{F}(\boldsymbol{z}) + \boldsymbol{z}, \tag{1.1.2}$$

the residual connection can be summarized as adding a *shortcut* connection to a composition of layers $\mathcal{F}$ (for example, the composition of convolutions and non-linear activations). While the initial motivation was to improve the optimization of deep networks, this procedure is at the base of a lot of the Unsupervised Domain Translation architectures [Zhu et al., 2017b, Choi et al., 2017, 2019, Taigman et al., 2017]. Without such architectural biases, those methods would most likely select a solution that does not preserve the structure of the source samples, as many solutions that minimize their objectives exist [de Bézenac et al., 2019] and could be selected. Furthermore, as demonstrated in chapter 2, the learned mapping does not preserve the semantics when the target images are not merely a different texture or colour than the source images.

1.1.2. Objectives

We are using this subsection to introduce several ideas that will be used in the subsequent chapters. Background on self-supervised learning objective, including contrastive learning objectives and BYOL, will be useful for both chapter 2 and chapter 3. Background on clustering and domain adaptation will be useful for chapter 2.

Contrastive learning. In contrastive learning, a bounded metric space is defined in which the aim is to both minimize the distance between the representation of a sample $\boldsymbol{z}_i = f_\theta(\boldsymbol{x}_i) : \boldsymbol{x} \in \mathcal{X}$ and the representation of a *positive* sample $\boldsymbol{z}_j = f_\theta(\boldsymbol{x}_j)$, and to maximize the distance between $\boldsymbol{z}_i$ and the representation of *negative* samples $f_\theta(\boldsymbol{x}') : \boldsymbol{x}' \in \mathcal{X} \setminus \boldsymbol{x}_i$. While the positive samples are typically augmented samples of $\boldsymbol{x}_i$, other strategies can be decided, such as choosing samples from the same labelled category [Khosla et al., 2021]. Noise contrastive estimation has been used by several methods [Chen et al., 2020b, He et al., 2020, Chen et al., 2020c, Hjelm et al., 2019a] and is defined as follows:

$$\mathcal{L}_{\text{nce}} := -\log \frac{\exp(d(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{\bar{\boldsymbol{x}} \in \mathcal{X} \setminus \boldsymbol{x}} \exp(d(\boldsymbol{z}_i, \bar{\boldsymbol{z}})/\tau)}, \tag{1.1.3}$$

where d is often taken to be the cosine similarity: $d(\boldsymbol{x}, \boldsymbol{y}) := \boldsymbol{x}^\top \boldsymbol{y} / ||\boldsymbol{x}||||\boldsymbol{y}||$.

BYOL.. Unlike most contrastive methods, BYOL [Grill et al., 2020] does not require negative samples. Instead, it introduces a target network in which the parameters ξ is a moving average of θ. They define the anchor and positive samples as $\boldsymbol{z_\theta} = f_\theta(t(\boldsymbol{x}))$ and $\boldsymbol{z_\xi} = f_\xi(t'((\boldsymbol{x}))$ respectively, where t,$t' \sim \mathcal{T}$ are augmentations sampled from a set of possible augmentations defined by the practitioner. Important to their method is a re-normalization of the representation using, for example, batch normalization [Ioffe and Szegedy, 2015], that acts as a repulsion force that prevents collapse and a stop-gradient operation on $\boldsymbol{z_\xi}$ that prevents the gradient from back-propagating through the target network. They also and introduce a *prediction* head and maps $\boldsymbol{z_\theta} \mapsto \boldsymbol{q_\theta}$. The objective to minimize is defined as follows:

$$\mathcal{L}_{\text{byol}} := 2 - d(\boldsymbol{q_\theta}, \boldsymbol{z_\xi}), \tag{1.1.4}$$

where d is explicitely defined as the cosine similarity.

The representation is typically evaluated on its ability to fit a linear classifier or cluster the samples on the learned representation. While no direct supervision is necessary during the training of the representation, the performance of the models trained with self-supervision is comparable to that obtained when training a classifier end-to-end with supervision. For example, for a comparable model, fitting a linear classifier on the representation learned with BYOL yields a score of 78.6% accuracy on ImageNet, while the supervised counterpart obtains a score of 78.9% [Grill et al., 2020]. The remarkable ability to fit a linear classifier on top of a representation of a model trained with self-supervision indicates that the representation has some structure for the data taken from the same training set distribution. However, as demonstrated in Figure 1, taken from Djolonga et al. [2021], the models trained with SimCLR are less robust to transfer learning than the models trained with supervision.

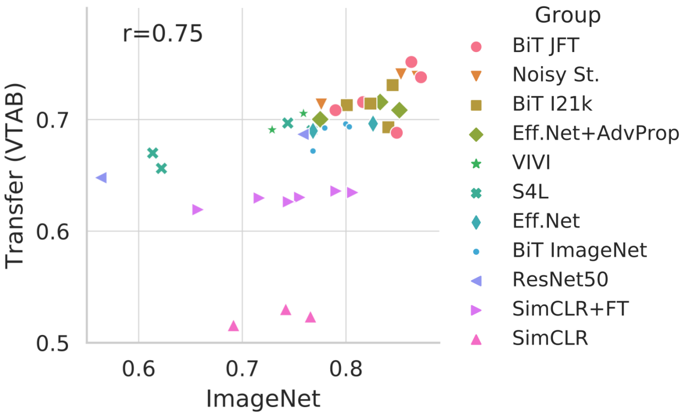


Fig. 1. Taken from Djolonga et al. [2021]. Demonstrate the transfer accuracy, using the VTAB transfer learning test suite [Zhai et al., 2019], of different learning methods and accuracy. We observe that SimCLR, the only self-supervised learning method, has a poor transfer accuracy given its ImageNet accuracy in comparison to the other training methods.

Clustering. The Clustering objective is to learn a grouping of the data. Commonly, clustering groups the data according to a similarity measure between the representation of the samples. Examples of algorithms implementing this idea include K-Means [Lloyd, 2006] and spectral clustering [Donath and Hoffman, 1973]. More recently, the idea to leverage the representation of a neural network to learn the clusters has started to emerge [Hu et al., 2017, Caron et al., 2018]. Deep cluster [Caron et al., 2018] propose to iteratively alternate between pseudo-labelling the data by applying k-means on the representation of a convolutional neural network and training the network using standard classification with the pseudo-labels. RIM [Gomes et al., 2010] and IMSAT [Hu et al., 2017] learn a mapping $c : \mathcal{X} \to \mathcal{C}$, where $\mathcal{C} \in \mathbb{R}^k$ is a continuous space representing a soft clustering of $\mathcal{X}$, by optimizing the following objective:

$$\min_c \lambda \mathcal{R}(c) - I(\mathcal{X};\mathcal{C}), \tag{1.1.5}$$

where $\lambda > 0$ is a Lagrange multiplier, I is the mutual information defined as

$$I(\mathcal{X};\mathcal{C}) = H(\mathcal{C}) - H(\mathcal{C}|\mathcal{X}) \tag{1.1.6}$$

The first term $H(\mathcal{C})$ encourages the samples to be distributed uniformly across the clusters, and the second term encourages the prediction to be confident (i.e. pushes the representation to be a one-hot vector). $\mathcal{R}$ of Equation 1.1.5 is a regularizer. For example, IMSAT uses the following regularizer:

$$\mathcal{R}(c) = \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_x}||c(\boldsymbol{x}) - c(\boldsymbol{x}')||_2^2, \tag{1.1.7}$$

where $\boldsymbol{x}' = t(\boldsymbol{x})$, and $t \sim \mathcal{T}$ is a set of transformations of the original image, such as affine transformations. Essentially, this approach is similar to the contrastive method presented above with the difference that we constraint the samples to cluster while pressuring them to be invariant under the set of transformations $\mathcal{T}$.

Clustering algorithms are usually evaluated on their ability to group the data according to some semantic labels defined externally, for example, by a human.

Unsupervised domain adaptation. Domain adaptation aims at adapting a function trained on a domain $\mathcal{X}$ with labelled samples to a domain $\mathcal{Y}$ with few or no labelled samples. Unsupervised domain adaptation refers to the case where the target domain $\mathcal{Y}$ is unlabelled during training.

Ben-David et al. [2010] shown that the error of a hypothesis function h on the target domain $\mathcal{Y}$ is upper bounded by the following

$$\mathcal{L}_{\mathcal{Y}}(h) \leq \mathcal{L}_{\mathcal{X}}(h) + d(\mathcal{X},\mathcal{Y}) + \min_{h'} \mathcal{L}_{\mathcal{X}}(h') + \mathcal{L}_{\mathcal{Y}}(h'), \tag{1.1.8}$$

where $\mathcal{L}_x$ is the error and can be computed given a loss function, for example the cross entropy and d is defined as the $\mathcal{H}$-divergence [Kifer et al., 2004].

We notice that the second term on the right-hand side of the bound of equation 1.1.8 is concerned with the divergence between two domains. To that end, the first approach to domain adaptation is to learn a representation of $\mathcal{X}$ and $\mathcal{Y}$ that minimizes the distance. Ganin et al. [2016] proposed the gradient reversal procedure that aims to match the marginal distribution of intermediate hidden representations of a neural network across domains. If h is a neural network and can be decomposed as $h = h_2 \circ h_1$, and assume that we have another parametric function D that serves as *domain discriminator*, then the gradient reversal objective to be minimized is defined as:

$$\mathcal{L}_{\text{gv}}(h_1, \mathcal{X}, \mathcal{Y}) = \max_{D} \ \mathbb{E}_{\boldsymbol{x} \sim \mathcal{X}}[\log D(h(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{y} \sim \mathcal{Y}}[\log(1 - D(h(\boldsymbol{y})))].$$

Minimizing this objective with respect to h_1 can also be seen as applying a GAN loss [Goodfellow et al., 2014] on a representation of a neural network effectively minimizing the discrepancy between $h_1(\mathcal{X})$ and $h_1(\mathcal{Y})$.

Minimizing the discrepancy between $h_1(\mathcal{X})$ and $h_1(\mathcal{Y})$ does not guarantee an alignment of the semantic categories of $\mathcal{X}$ and $\mathcal{Y}$ at the representation level. For example, assuming that $\mathcal{X}$ and $\mathcal{Y}$ have two label categories. Then, the representation $h_1(\mathcal{X})$ with $label = 0$ could overlap with the representation $h_1(\mathcal{Y})$ with $label = 1$ and vice versa. To counteract this problem, Shu et al. [2018] propose to regularize the training via the cluster assumption. The cluster assumption [Chapelle and Zien, 2005] is simply an assumption that the data is clusterable into classes. In other words, it states that the decision boundaries of h_2 should be in low-density regions of the data. One way to push forward this assumption is to encourage the representation of both $\mathcal{X}$ and $\mathcal{Y}$ to clusters into dense regions. Therefore, assuming that h_2 is relatively low capacity (for example a linear classifier), losing alignment during training would imply that the decision boundary of h_2 would traverse a dense region of $h_1(\mathcal{X})$ of $h_1(\mathcal{Y})$. To enforce the cluster assumption, Grandvalet and Bengio [2005] propose to minimize the following objective on the conditional entropy of a prediction given a sample:

$$\mathcal{L}_c(h, \mathbb{P}_y) = -\mathbb{E}_{\boldsymbol{y} \sim \mathbb{P}_y} h(\boldsymbol{y}) \log h(\boldsymbol{y}).$$

However, such a constraint is applied to an empirical distribution in practice. Hence, nothing stops the classifier from abruptly changing its predictions for any samples outside of the training distribution. This motivates the following constraints. Virtual adversarial training [Shu et al., 2018] proposes to alleviate this problem by constraining h to be locally-Lipschitz around an ϵ-ball. Borrowing from Miyato et al. [2018], they propose the additional regularizer to encourage local-lipschitzness:

$$\mathcal{L}_v(h, \mathbb{P}) = \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}} \max_{||\boldsymbol{r}||_2 \leq \epsilon} D_{\mathrm{KL}}(h(\boldsymbol{x}) \,||\, h(\boldsymbol{x} + \boldsymbol{r})),$$

with $\epsilon > 0$. Virtual mixup training [Mao et al., 2019], with a similar motivation, proposes that the prediction of an interpolated point $\tilde{\boldsymbol{x}}$ should itself be an interpolation of the predictions at $\boldsymbol{x}_1$ and at $\boldsymbol{x}_2$. The interpolated samples are defined as

$$\tilde{\boldsymbol{x}} = \alpha \boldsymbol{x}_1 + (1 - \alpha) \boldsymbol{x}_2,$$

$$\tilde{\boldsymbol{y}} = \alpha h(\boldsymbol{x}_1) + (1 - \alpha) h(\boldsymbol{x}_2),$$

with $\alpha \sim U(0{,}1)$, where $U(0{,}1)$ is a continuous uniform distribution between 0 and 1.

The proposed objective is then simply

$$\mathcal{L}_m(h, \mathbb{P}) = -\mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}) \sim \mathbb{P}} \tilde{\boldsymbol{y}}^\top \log h(\tilde{\boldsymbol{x}}).$$

These objectives are composed to give the overall optimization problem:

$$\min_h \mathcal{L}_{\mathcal{X}}(h, \mathbb{P}_x) + \lambda_1 \mathcal{L}_{\mathrm{gv}}(h_1, \mathbb{P}_x, \mathbb{P}_y) + \lambda_2 \mathcal{L}_{\mathrm{c}}(h, \mathbb{P}_y) +$$

$$\lambda_3 \mathcal{L}_{\mathrm{v}}(h, \mathbb{P}_x) + \lambda_4 \mathcal{L}_{\mathrm{v}}(h, \mathbb{P}_y) + \lambda_5 \mathcal{L}_{\mathrm{m}}(h, \mathbb{P}_x) + \lambda_6 \mathcal{L}_{\mathrm{m}}(h, \mathbb{P}_y).$$

The pressure from the gradient reversal and the objective implementing the cluster assumption, combined with the architectural bias, encourage the representation to cluster the samples according to their semantic categories.

In the next chapter, we will combine the ideas presented in this section to learn domain invariant clusters without supervision to improve the Unsupervised Domain Translation methods. Next, we review some idea from the community of artificial language emergence that will be useful for chapter 3.

1.2. Emergence of structure in artificial languages

Language has powerful properties that enable us to communicate and understand novel ideas. Assuming that a speaker and a listener agree on syntactic and grammatical rules, they can exchange and build knowledge of things they have never seen or even imagined before. This capacity may be attributed to the compositional nature of language: the semantics of a complex sentence is entirely determined by the semantics of its constituents and the syntactic rules. For example, the nominal sentence "the blue cat" is entirely determined by the syntactic rules of English and the semantics of the words "the," "blue," and "cat." We can further compose "The blue cat" with another nominal sentence – e.g., "the red cat" – and conjunction to define a more complex idea, which can be further combined. Therefore, someone that knows a set of semantical elements can recursively and combinatorially combine them to obtain a countably infinite number of complex semantical elements.

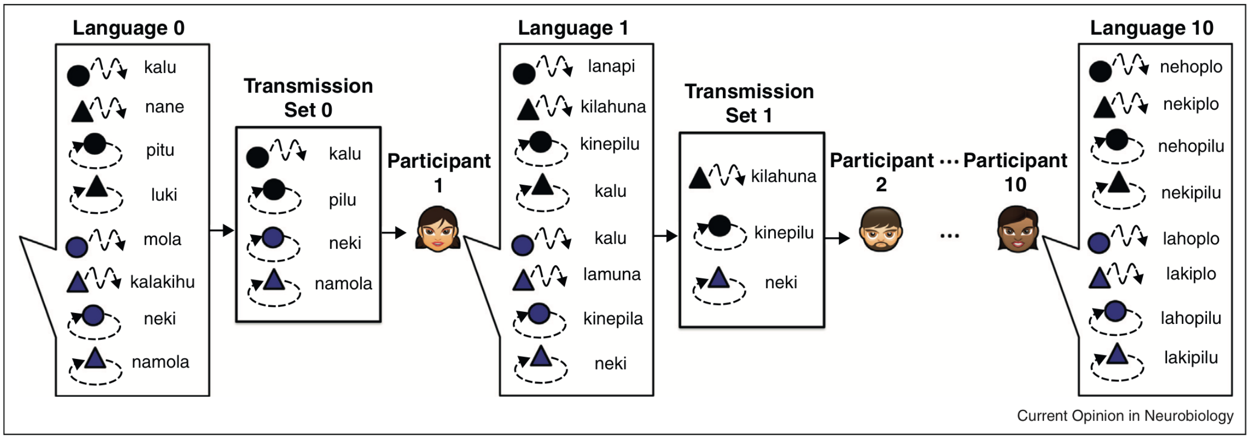
Another attractive property of language is that the capacity to understand a complex sentence implies the capacity to understand structurally related sentences. For example, if someone can understand the nominal sentences "the blue cat" and "the red dog," then they should be able to understand the nominal sentences "the red cat" and "the blue dog." This property, called systematicity, comes from the fact that the sentences we understand share structure due to the syntactical rules. We note that this property is closely related to out-of-distribution generalization. The fact that we can understand structurally related complex sentences without encountering them a priori is a property of generalization that we would want our models to enjoy. We take inspiration from the language emergence literature to explore how such properties could emerge and whether we can set the same condition in a representation learning framework.

The origin of language and why it has its structural properties is still up for debate. Many theories exist for describing the emergence of the structure in language, ranging from being innate [Chomsky, 1959] to having emerged due to several pressures and constraints [Kirby and Hurford, 2002, Christiansen and Chater, 2008]. This document entertains the later idea and hypothesizes that compositional language emerges given the right external conditions.

Artificial language emergence is typically framed as the solution to a coordination problem among N agents [Lewis, 1969]. The optimal language that emerges between the agents is merely a Nash equilibrium of the optimization problem that maximizes the agents' utility when both of them have common interests and are incentivized to coordinate and communicate. However, such language only needs to serve the purpose of the agents and does not need to be compositional. In other words, without more constraints, an optimal language may be an idiosyncratic language that correctly allows the agent to signal their intent but does not generalize to new contexts. For example, Vervet Monkey's language for signalling the presence of a predator is idiosyncratic in the sense that it has a different sound for the different predators [Oliphant and Batali, 1997]. Their language for signalling the presence of a predator has not reached a point where it is compositional, perhaps because it is not needed to serve its purpose.

Kirby and Hurford [2002] suggest that compositionality in natural languages emerged in part due to a biological pressure since compositionality can improve organization and efficiency, which in turn can help with survival and reproduction. But, also due to cultural transmission, the transmission from generation to generation of a growing and evolving language. The pressure to transmit a rich set of meanings, with compressive constraints due to having to propagate it to new generations, while preserving the expressivity of the language could have pressured the language to develop a compositional structure. Christiansen and Chater [2008] propose that language emerged solely due to a set of constraints that could have biased the language to have a compositional structure through cultural evolution. Some constraints include the number of semantical elements that we can remember and perceptuo-motors constraints.

The expressivity of a language relates to the number of different signals available to describe a set of meanings. For example, a language is expressive if each meaning has its own signal and is degenerate – we call it collapsed in the ML literature – if all the meaning maps to a single signal. Kirby et al. [2015] demonstrate, in a simulation of artificial language and a clinical trial, that having both an objective to preserve the expressivity of the language and a compressive prior leads to compositionality in the language. However, having only one pressure leads to undesirable effects. Solely a compressive pressure leads to a degenerate language. An expressive pressure exclusively leads to a holistic language where each meaning maps to an idiosyncratic signal. One can see that a compositional language has a vocabulary



Current Opinion in Neurobiology

Fig. 2. Taken from: . Illustration of the Iterated Learning Model. The subset of a holistic language describing a set of sketches at iteration 0 is being transmitted to the first agent which learns it and in turns transmits a subset of it learned language to the next agent. This procedure is iterated until a compositional language emerges. At the last iteration, we observe a structured language that describe the color, the shape and the line's pattern.

that is exponentially smaller for the same expressivity that a holistic language. Assuming that the language is fully compositional, each semantic element can be combined with the other semantic elements to produce a distinct meaning. Therefore, assuming that we have a composition of L semantic elements taken from a set of V possible elements, then the number of meaning is upper bounded by $\Omega(V^L)$. In comparison, the number of meanings in a holistic language scale in $O(V)$. While an emergent language is generally not fully compositional, nor fully holistic [Kirby et al., 2015], this argument demonstrates the relation between compressibility, expressivity and compositionality.

Kirby et al. [2014] proposed the Iterated Learning Model as a way to simulate cultural transmission and to study this information bottleneck hypothesis. The idea is that compositional language can emerge from holistic and unstructured language if the language is repeatedly transmitted via a restricted set of utterances to new agents oblivious to the language. This transmission to new agents induces a compressive pressure that leads to compositionality in a language, as demonstrated in several experiments [Kirby et al., 2014, Kirby and Hurford, 2002, Ren et al., 2020b]. We depict a sketch of the learning dynamics in Figure 2 where we see the evolution of a language describing a set of sketches. As the number of iteration grows, the language evolves from being holistic to being compositional, where the different attributes are encoded by a set of coherent letters at specific locations. In other words, we can observe a set of semantic elements – describing the attributes – with syntactic rules describing the observation as a whole.

The Iterated Learning Model can be applied in different type of communication games. We will review how cultural transmission can induce structure in the Signaling game [Oliphant and Batali, 1997, Lewis, 1969].

1.2.1. Communication games

Communications games are multi-agents games. This document will consider games between only two agents for simplicity. The two agents are a sender f and a receiver g. The training setup involves the sender signalling a message to the receiver by mapping from a set of *meaning*, $\mathcal{X}$, to a set of *signals*, $\mathcal{Z}$. The receiver performs some task using either $\mathcal{X}$, $\mathcal{Z}$ or both depending on the game.

Object selection game. In the object selection game, a sender is given an object to encode into a discrete signal $f_\theta : \mathcal{X} \rightarrow \mathcal{Z}$. The object is often defined as a symbolic input representing the attributes or, in some cases, as an image [Lazaridou et al., 2018, Choi et al., 2018]. The receiver receives the message $\boldsymbol{z}$ as well as a set of N objects $[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$ and has to choose the same object that the sender received and communicated to the receiver. In other words, the receiver maps $g_{\psi_1} : \mathcal{Z} \rightarrow \mathbb{R}^d$ and maps $g_{\psi_2} : \mathcal{X} \rightarrow \mathbb{R}^d$. An embedding $(c_i)_{i=1}^n$, where $c_i := \arg\max g_{\psi_1}(\boldsymbol{z}) \cdot g_{\psi_2}(\boldsymbol{x}_i)^\top$ is generated for each sample given to the receiver. Both agents are rewarded if the receiver correctly choose the right object c' that was given to the sender, which create an incentive to collaborate [Das et al., 2017, Kottur et al., 2017, Ren et al., 2019]. We can define $p_f(\boldsymbol{z}|\boldsymbol{x})$ the policy of the sender to select a message z and $p_g(\bar{c}|\boldsymbol{c}_1,...,c_n)$ the policy of the receiver to select the right object. [Ren et al., 2019] defines the following update rules for the sender and the receiver using REINFORCE [Williams, 1992]:

$$\nabla_\theta J := E\left[R(\bar{c},\boldsymbol{x})\nabla_\theta \log p_f(\boldsymbol{z}|\boldsymbol{x}) + \lambda_f \nabla_\theta H[p_f(\boldsymbol{z}|\boldsymbol{x})]\right], \tag{1.2.1}$$

$$\nabla_\psi J := E\left[R(\bar{c},\boldsymbol{x})\nabla_\psi \log p_g(\bar{c}|\boldsymbol{z},c_1,...,c_k) + \lambda_g \nabla_\psi H(p_g(\bar{c}|\boldsymbol{x}, c_1,..., c_n))\right]. \tag{1.2.2}$$

Ren et al. [2019] proposed an iterated learning version of this game. An iteration is divided into three phases: The interaction phase, the transmission phase and the learning phase. The interaction phase corresponds to the selection game presented above. The transmission phase corresponds to generating a set of N pairs from a subset of the training set:

$$\mathcal{Z} := \{(\boldsymbol{x}_i, \boldsymbol{z}_i)\}_{i=1}^{N}, \tag{1.2.3}$$

where $z_i \sim p_f(\boldsymbol{z}|\boldsymbol{x}_i)$. $\mathcal{Z}$ is then used in the learning phase. The learning phase is used to pre-train a newly initialized sender $f_{\theta^{t+1}}$ to imitate the generated messages of the previous sender given an observation. This is done as follows:

$$\min_{\theta^{t+1}} E_{(\boldsymbol{x},\boldsymbol{z})\sim\mathcal{Z}} l(f_{\theta^{t+1}}(\boldsymbol{x}), \boldsymbol{z}), \tag{1.2.4}$$

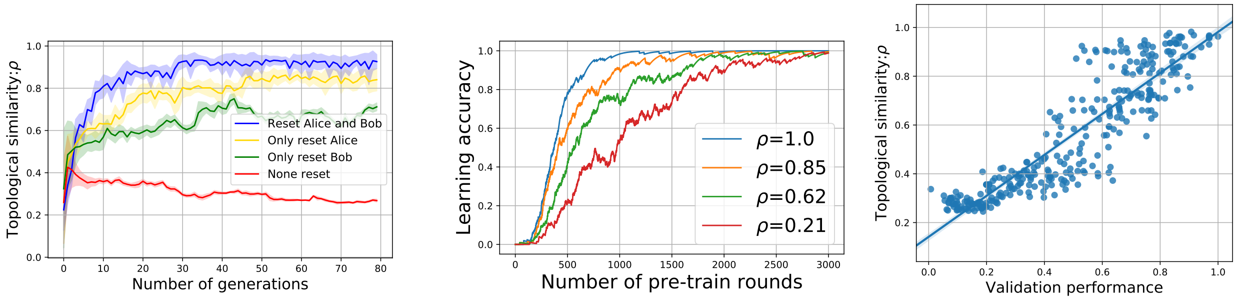
where l is the cross-entropy. The transmission and the training phase procedures are similar to the distillation procedure Hinton et al. [2015] used to train a network in machine learning. The difference is that $\boldsymbol{z}$ is a discrete sample from a categorical distribution $p(\boldsymbol{z}|\boldsymbol{x})$, rather than a single soft-target. In summary, the Neural Iterated Learning procedure trains a sender and a receiver using a communication game followed by a noisy hard-distillation procedure to pre-train a new sender for the next iteration of the communication game. Algorithm 1 depicts this procedure.

Algorithm 1 Neural Iterated Learning

```
Require: $\mathcal{X}$, $f_{\theta^0}$, $g_{\phi^0}$, $N_{\text{iter}}$, $M_{\text{interaction}}$.
  $\theta^0$ randomly initialized
  $\phi^0$ randomly initialized
  $t \leftarrow 0$
  while $N_{\text{iter}} \neq 0$ do
      $S \leftarrow 0$
      while $S \neq M$ do
          $\theta^t \leftarrow \theta^t + \alpha \nabla_{\theta^t} J$                 ▷ Update using equation 1.2.1.
          $\psi^t \leftarrow \psi^t + \alpha \nabla_{\psi^t} J$                 ▷ Update using equation 1.2.2.
          $S \leftarrow S + 1$
      end while
      $\mathcal{Z} \leftarrow$ Generation($\mathcal{X}$, $f_{\theta^t}$)              ▷ Generate using equation 1.2.3.
      $\theta^{t+1} \leftarrow$ Distillation($\mathcal{Z}$, $f_{\theta^{t+1}}$)          ▷ Distill using equation 1.2.4.
      $\psi^{t+1}$ randomly initialized
      $t \leftarrow t + 1$
      $N_{\text{iter}} \leftarrow N_{\text{iter}} - 1$
  end while
```

Topographical similarity. The topographical similarity (ρ) is a measure that captures how much a message $\boldsymbol{z}$ retains the structure of the original object description $\boldsymbol{x}$. It is defined as the negative Spearman's correlation between the pairwise cosine similarities of object vectors and the pairwise Levenshtein distances between all messages. We take the negative Spearman's correlation since we correlate distances and similarities. Hence, a topographical similarity of 1 indicates that the message preserves the object description structure and indicates a compositional language.

Using the topographical similarity, Ren et al. [2019] demonstrated that resetting the sender, the receiver or both increased the structure of the generated messages, as demonstrated in Figure 3a. Their hypothesis is that language with a topographical similarity should be easier to learn and thus more likely to be selected than a language with a small topographical similarity. They empirically demonstrate their claim as reproduced in Figure 3b



(a) **(b)** **(c)**

Fig. 3. Figures taken from [Ren et al., 2019]. (a) Topographical similarity (also called topological similarity) given which of the sender (Alice) and/or the receiver (Bob) gets resetted and retrained. (b) The training accuracy given the topographical similarity of the message of the sender. (c) Relationship between the topographical similarity and the validation performance on the object selection game.

where they demonstrate the learning speed given the accuracy of the agents on the object selection game and observe that a lower topographical similarity induces a faster learning speed.

They also demonstrate a linear relationship between the performance in the object selection game on unseen samples and the topographical similarity, shown in Figure 3c. The idea that a more structured representation induces increased generalization is interesting and of general interest to the representation learning community.

We now turn to our first contribution on learning domain invariant representation without supervision and its application on Unsupervised Domain Translation. While the ideas presented in this Section are not closely related to the content of the next chapter, they will be in the subsequent chapters.

Chapter 2

Integrating Categorical Semantics into Unsupervised Domain Translation

Note: *This chapter has been published as an article to the Ninth International Conference on Learning Representations (ICLR 2021) with the following co-authors: Samuel Lavoie, Faruk Ahmed and Aaron Courville.*

2.1. Introduction

Domain translation has sparked a lot of interest in the computer vision community following the work of Isola et al. (2016) on *image-to-image translation*. This was done by learning a conditional GAN [Mirza and Osindero, 2014], in a supervised manner, using paired samples from the source and target domains. CycleGAN [Zhu et al., 2017a] considered the task of unpaired and unsupervised image-to-image translation, showing that such a translation was possible by simply learning a mapping and its inverse under a *cycle-consistency* constraint, with GAN losses for each domain.

But, as has been noted, despite the cycle-consistency constraint, the proposed translation problem is fundamentally ill-posed and can consequently result in arbitrary mappings [Benaim et al., 2018, Galanti et al., 2018, de Bézenac et al., 2019]. Nevertheless, CycleGAN and its derivatives have shown impressive empirical results on a variety of image translation tasks. Galanti et al. [2018] and de Bézenac et al. [2019] argue that CycleGAN's success is owed, for the most part, to architectural choices that induce implicit biases toward *minimal complexity* mappings. That being said, CycleGAN, and follow-up works on unsupervised domain translation, have commonly been applied on domains in which a translation entails little geometric changes and the style of the generated sample is independent of the semantic content in the source sample. Commonly showcased examples include translating edges↔shoes and horses↔zebras.

While these approaches are not without applications, we demonstrate two situations where unsupervised domain translation methods are currently lacking. The first one, which we call *Semantic-Preserving Unsupervised Domain Translation* (SPUDT), is defined as translating, without supervision, between domains that share common semantic attributes. Such attributes may be a non-trivial composition of features obfuscated by domain-dependent spurious features, making it hard for the current methods to translate the samples while preserving the shared semantics despite the implicit bias. Translating between MNIST↔SVHN is an example of translation where the shared semantics, the digit identity, is obfuscated by many spurious features, such as colours and background distractors, in the SVHN domains. In section 2.4.1, we take this specific example and demonstrate that using domain invariant categorical semantics improves the digit preservation in UDT.

The second situation that we consider is *Style-Heterogeneous Domain Translation* (SHDT). SHDT refers to a translation in which the target domain includes many semantic categories, with a distinct style per semantic category. We demonstrate that, in this situation, the style encoder must be conditioned on the shared semantics to generate a style consistent with the semantics of the given source image. In Section 2.4.2, we consider an example of this problem where we translate an ensemble of sketches, with different objects among them, to real images.

In this paper, we explore both the SPUDT and SHDT settings. In particular, we demonstrate how domain invariant categorical semantics can improve translation in these settings. Existing works [Hoffman et al., 2018, Bousmalis et al., 2017] have considered semi-supervised variants by training a classifier with labels on the source domain. But, differently from them, we show that it is possible to perform well at both kinds of tasks *without any supervision*, simply with access to unlabelled samples from the two domains. This additional constraint may further enable applications of domain translation in situations where labelled data is absent or scarce.

To tackle these problems, we propose a method which we refer to as Categorical Semantics Unsupervised Domain Translation (CatS-UDT). CatS-UDT consists of two steps: (1) learning an inference model of the shared categorical semantics across the domains of interest without supervision and (2) using a domain translation model in which we condition the style generation by inferring the learned semantics of the source sample using the model learned at the previous step. We depict the first step in Figure 1b and the second in Figure 2.

More specifically, the contributions of this work are the following:

- Novel framework for learning invariant categorical semantics across domains (Section 2.3.1).
- Introduction of a method of semantic style modulation to make SHDT generations more consistent (Section 2.3.2).

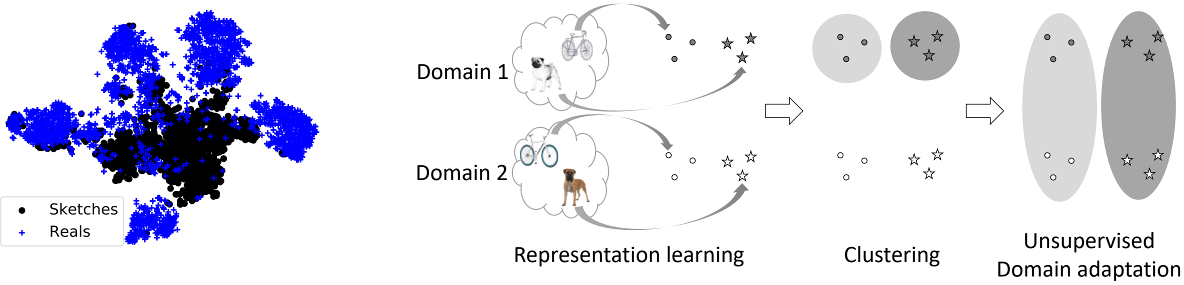
- Comparison with UDT baselines on SPUDT and SHDT highlighting their existing challenges and demonstrating the relevance of our incorporating semantics into UDT (Section 2.4).

2.2. Related works

Domain translation is concerned with translating samples from a source domain to a target domain. In general, we categorize a translation that uses pairing or supervision through labels as *supervised domain translation* and a translation that does not use pairing or labels as *unsupervised domain translation*.

Supervised domain translation methods have generally achieved success through either the use of pairing or the use of supervised labels. Methods that leverage the use of category labels include Taigman et al. [2017], Hoffman et al. [2018], Bousmalis et al. [2017]. The differences between these approaches lie in particular architectural choices and auxiliary objectives for training the translation network. Alternatively, Isola et al. [2016], Gonzalez-Garcia et al. [2018], Wang et al. [2018, 2019], Zhang et al. [2020] leverage paired samples as a signal to guide the translation. Also, some works propose to leverage a segmentation mask [Tomei et al., 2019, Roy et al., 2019, Mo et al., 2019]. Another strategy is to use the representation of a pre-trained network as semantic information [Ma et al., 2019, Wang et al., 2019, Wu et al., 2019, Zhang et al., 2020]. Such a representation typically comes from the intermediate layer of a VGG [Liu and Deng, 2015] network pre-trained with labelled ImageNET [Deng et al., 2009]. Conversely to our work, [Murez et al., 2018] propose to use image-to-image translation to regularize domain adaptation.

Unsupervised domain translation considers the task of domain translation without any supervision, whether through labels or pairing of images across domains. CycleGAN [Zhu



(a) ResNET-50 trained using MOCO.

(b) Domain invariant categorial representation learning.

Fig. 1. (a) T-SNE embeddings of the representation of Sketches and Reals taken from a hidden layer for a pre-trained model on ImageNET, (b) Sketch of our method for learning a domain invariant categorial semantics.

et al., 2017a] proposed to learn a mapping and its inverse constrained with a *cycle-consistency* loss. The authors demonstrated that CycleGAN works surprisingly well for some translation problems. Later works have improved this class of models [Liu et al., 2017, Kim et al., 2017, Almahairi et al., 2018, Huang et al., 2018, Choi et al., 2017, 2019, Press et al., 2019], enabling multi-modal and more diverse generations. But, as shown in Galanti et al. [2018], the success of these methods is mostly due to architectural constraints and regularizers that implicitly bias the translation toward mappings with minimum complexity. We recognize the usefulness of this inductive bias for preserving low-level features like the pose of the source image. This observation motivates the method proposed in Section 2.3.2 for conditioning the style using the semantics.

2.3. Categorical Semantics Unsupervised Domain translation

In this section, we present our two main technical contributions. First, we discuss an unsupervised approach for learning categorical semantics that is invariant across domains.

Next, we incorporate the learned categorical semantics into the domain translation pipeline by conditioning the style generation on the learned categorical-code.

2.3.1. Unsupervised learning of domain invariant categorical semantics

The framework for learning the domain invariant categorical representation, summarized in Figure 1b, is composed of three constituents: unsupervised representation learning, clustering and domain adaptation. First, embed the data of the source and target domains into a representation that lends itself to clustering. This step can be ignored if the raw data is already in a form that can easily be clustered. Second, cluster the embedding of one of the domains. Third, use the learned clusters as the ground truth label in an unsupervised domain adaptation method. We provide more details on the methods used in Section 2.4. Here, we motivate the utility of each of the constituents and describe how they are used in the present framework.

Representation learning. Pre-trained supervised representations have been used in many instances as a way to preserve alignment in domain translation [Ma et al., 2019, Wang et al., 2019]. In contrast to prior works that use models trained with supervision, we use models trained with self-supervision [van den Oord et al., 2018a, Hjelm et al., 2019a, He et al., 2020, Chen et al., 2020b]. Self-supervision defines objectives that depends only on the intrinsic information within data. This allows for the use of unlabelled data, which in turn could enable the applicability of domain translation to modalities or domains where labelled data is scarce. In this work, we consider the noise contrastive estimation [van den Oord et al., 2018b] which minimizes the distance in a normalized representation space between an anchor sample and its transformation and maximizes the distance between the same anchor sample and another sample in the data distribution. Formally, we learn the embedding function $d : \mathcal{X} \to \mathbb{R}^D$ of samples $\boldsymbol{x} \in \mathcal{X}$ as follows:

$$\arg\min_d -\mathbb{E}_{\boldsymbol{x}_i \sim \mathcal{X}} \log \frac{\exp(d(\boldsymbol{x}_i) \cdot d(\boldsymbol{x}_i')/\tau)}{\sum_{j=0}^{K} \exp(d(\boldsymbol{x}_i) \cdot d(\boldsymbol{x}_j)/\tau)}, \tag{2.3.1}$$

where $\tau > 0$ is a hyper-parameter, $\boldsymbol{x}_i$ is the anchor sample with its transformation $\boldsymbol{x}_i' = t(\boldsymbol{x}_i)$ and $t : \mathcal{X} \to \mathcal{X}$ defines the set of transformations that we want our embedding space to be invariant to.

While other works use the learned representation directly in the domain translation model, we propose to use it as a leverage to obtain a categorical and domain invariant embedding as described next. In some instances, the data representation is already amenable to clustering. In those cases, this step of representation learning can be ignored.

Clustering allows us to learn a categorical representation of our data without supervision. Some advantages of using such a representation are as follows:

- A categorical representation provides a way to select exemplars without supervision by simply selecting an exemplar from the same categorical distribution of the source sample.
- The representation is straightforward to evaluate and to interpret. Samples with the same semantic attributes should have the same cluster.

In practice, we cluster one domain because, as we see in Figure 1a, the continuous embedding of each domain obtained from a learned model may be disjoint when they are sufficiently different. Therefore, a clustering algorithm would segregate each domain into its own clusters. Also, the domain used to determine the initial clusters is important as some domains may be more amenable to clustering than others. Deciding which domain to cluster depends on the data and the choice should be made after evaluation of the clusters or inspection of the data.

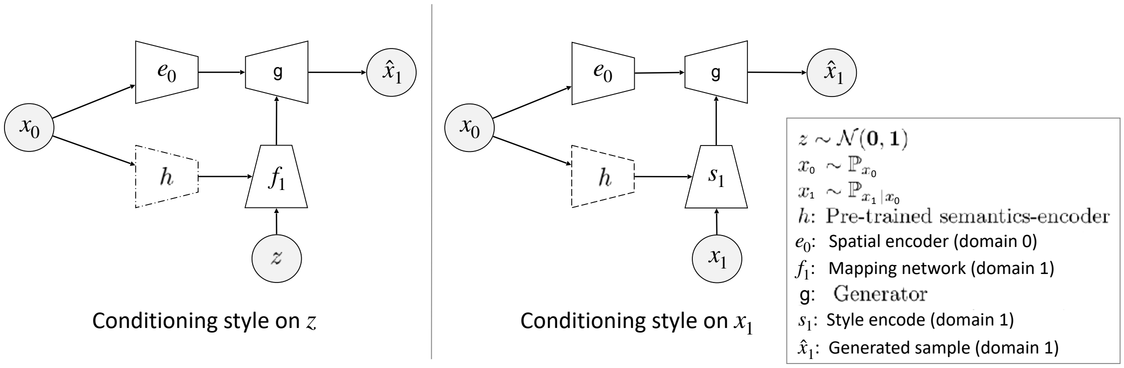
More formally, consider $\mathcal{X}_0 \subset \mathbb{R}^N$ be the domain chosen to be clustered. Assume a given embedding function $d : \mathcal{X}_0 \to \mathbb{R}^D$ that can be learned using self-supervision. If $\mathcal{X}_0$ is already cluster-able, d can be the identity function. Let $c : \mathbb{R}^D \to \mathcal{C}$ be a mapping from the embedding of $\mathcal{X}_0$ to the space of clusters $\mathcal{C}$. We propose to cluster the embedding representation of the data:

$$\arg\min_{c} \mathrm{C}(c,\, d(\mathcal{X}_0)), \tag{2.3.2}$$

where C is a clustering objective. The framework is agnostic to the clustering algorithm used. In our experiments (Section 2.4), we considered IMSAT [Hu et al., 2017] for clustering MNIST and Spectral Clustering [Donath and Hoffman, 1973] for clustering the learned embedding of our real images.

Unsupervised domain adaptation. Given clusters learned using samples from a domain $\mathcal{X}_0$, it is unlikely that such clusters will generalize to samples from a different domain with a considerable shift. This can be observed in Figure 1a where, if we clustered the samples of the real images, it is not clear that the samples from the Sketches domain would semantically cluster as we expect. That is, samples with the same semantic category may not be grouped in the same cluster.

Unsupervised domain adaptation [Ben-David et al., 2010] tries to solve this problem where one has one supervised domain. However, rather than using labels obtained through supervision from a source domain, we propose to use the learned clusters as ground-truth labels on the source domain. This modification allows us to adapt and make the clusters learned on one domain invariant to the other domain.



e_0
g
$\hat{x}_1$
x_0
h
f_1
z
Conditioning style on z
e_0
g
$\hat{x}_1$
x_0
h
s_1
x_1
Conditioning style on x_1
$z \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$
$x_0 \sim \mathbb{P}_{x_0}$
$x_1 \sim \mathbb{P}_{x_1|x_0}$
h: Pre-trained semantics-encoder
e_0: Spatial encoder (domain 0)
f_1: Mapping network (domain 1)
g: Generator
s_1: Style encode (domain 1)
$\hat{x}_1$: Generated sample (domain 1)

Fig. 2. Our proposed adaptation to the image-to-image framework for CatS-UDT. **Left**: generate the style using a mapping network conditioned on both noise $z \sim \mathcal{N}(0,1)$ and the semantics of the source sample $h(\boldsymbol{x}_0)$. **Right**: infer style of an exemplar $\boldsymbol{x}_2$ using a style encoder and $h(\boldsymbol{x}_0)$.

More formally, given two spaces $\mathcal{X}_0 \in \mathbb{R}^N$, $\mathcal{X}_1 \in \mathbb{R}^N$ representing the data space of domains 0 and 1 respectively, given a C-way one-hot mapping of the embedding of domain 0 to clusters, $c : d(\mathcal{X}_0) \to \mathcal{C}$ ($\mathcal{C} \subset \{0,1\}^C$), we propose to learn an adapted clustering $h : \mathcal{X}_0 \cup \mathcal{X}_1 \to \mathcal{C}$. We do so by optimizing:

$$\arg\min_h -\mathbb{E}_{\boldsymbol{x}_0 \sim \mathcal{X}_0} c(d(\boldsymbol{x}_0)) \log h(\boldsymbol{x}_0) + \Omega(h, \mathcal{X}_0, \mathcal{X}_1). \tag{2.3.3}$$

Ω represents the regularizers used in unsupervised domain adaptation. The framework is also agnostic to the regularizers used in practice. In our experiments, the regularizers comprised of gradient reversal [Ganin et al., 2016], VADA [Shu et al., 2018] and VMT [Mao et al., 2019]. We describe those regularizers in more detail in Section 1.1.2.

2.3.2. Conditioning the style encoder of Unsupervised Domain Translation

Recent methods for unsupervised image-to-image translation have two particular assets: (1) they can work with few training examples, and (2) they can preserve spatial coherence such as pose. With that in mind, our proposition to incorporate semantics into UDT, as depicted in figure 2, is to incorporate semantic-conditioning into the style inference of a domain translation framework. We will consider that the semantics is given by a network (h in Figure 2). The rationale behind this proposition originates from the conclusions by Galanti et al. [2018], de Bézenac et al. [2019] that the unsupervised domain translation methods work due to an inductive bias toward minimum complexity mappings. By conditioning only the style encoder on the semantics, we preserve the same inductive bias in the spatial encoder, forcing the generated sample to preserve some spatial attributes of the source sample, such as pose, while conditioning its style on the semantics of the source sample. In practice, we can learn the domain invariant categorical semantics, without supervision, using the method described in the previous subsection.

There can be multiple ways for incorporating the style into the translation framework. In this work, we follow an approach similar to the one used in StyleGAN [Karras et al., 2019] and StarGAN-V2 [Choi et al., 2019]. We incorporate the style, conditioned on the semantics, by modulating the latent feature maps of the generator using an Adaptive Instance Norm (AdaIN) module [Huang and Belongie, 2017]. Next, we describe each network used in our domain translation model and the training of the domain translation network.

2.3.2.1. Networks and their functions.

Content encoders, denoted e, extract the spatial content of an image. It does so by encoding an image, down-sampling it to a representation of resolution smaller or equal than the initial image, but greater than one to preserve spatial coherence.

Semantics encoder, denoted h, extracts semantic information defined as a categorical label. In our experiments, the semantics encoder is a pre-trained network.

Mapping networks, denoted f, encode $\boldsymbol{z} \sim \mathcal{N}(0{,}1)$ and the semantics of the source image to a vector representing the style. This vector is used to condition the AdaIN module used in the generator which modulates the style of the target image.

Style encoders, denoted s, extract the style of an exemplar image in the target domain. This style is then used to modulate the feature maps of the generator using AdaIN.

Generator, denoted g, generates an image in the target domain given the content and the style. The generator upsamples the content, injecting the style by modulating each layer using an AdaIN module.

2.3.2.2. Training. Let $\boldsymbol{x}_0 \sim \mathbb{P}_{x_0}$ and $\boldsymbol{x}_1 \sim \mathbb{P}_{x_1}$ be samples from two probability distributions on the spaces of our two domains of interest. Let $\boldsymbol{z} \sim \mathcal{N}(0, 1)$ samples from a Gaussian distribution. Let $y \sim \mathcal{B}(0.5)$ defines the domain, sampled from a Bernoulli distribution, and its inverse $\bar{y} := 1 - y$. We define the following objectives for samples generated with the mapping networks f and the style encoder s:

Adversarial loss [Goodfellow et al., 2014]. Constrain the translation network to generate samples in distribution to the domains. Consider $d_{\cdot}$ the discriminators [1].

$$\begin{aligned} \mathcal{L}^{f}_{\text{adv}} &:= \mathbb{E}_y \left[\mathbb{E}_{\boldsymbol{x}_{\bar{y}}} \log d^{f}_{\bar{y}}(x_{\bar{y}}) + \mathbb{E}_{\boldsymbol{x}_y} \mathbb{E}_{\boldsymbol{z}} \log(1 - d^{f}_{\bar{y}}(g(e_y(\boldsymbol{x}_y),\, f_{\bar{y}}(h(\boldsymbol{x}_y),\, \boldsymbol{z})))) \right], \\ \mathcal{L}^{s}_{\text{adv}} &:= \mathbb{E}_y \left[\mathbb{E}_{\boldsymbol{x}_{\bar{y}}} \log d^{s}_{\bar{y}}(x_{\bar{y}}) + \mathbb{E}_{\boldsymbol{x}_y} \mathbb{E}_{\boldsymbol{x}_{\bar{y}} \sim \mathbb{P}_{x_{\bar{y}} | h(x_y)}} \log(1 - d^{s}_{\bar{y}}(g(e_y(\boldsymbol{x}_y),\, s_{\bar{y}}(h(\boldsymbol{x}_y),\, \boldsymbol{x}_{\bar{y}})))) \right]. \end{aligned} \tag{2.3.4}$$

Cycle-consistency loss [Zhu et al., 2017a]. Regularizes the content encoder and the generator by enforcing the translation network to reconstruct the source sample.

$$\begin{aligned} \mathcal{L}^{f}_{\text{cyc}} &:= \mathbb{E}_y \left[\mathbb{E}_{\boldsymbol{x}_y} \mathbb{E}_{\boldsymbol{z}} \left| \boldsymbol{x}_y - g(e_{\bar{y}}(g(e_y(\boldsymbol{x}_y),\, f_{\bar{y}}(h(\boldsymbol{x}_1),\, \boldsymbol{z}))),\, s_y(h(\boldsymbol{x}_y),\, \boldsymbol{x}_y)) \right|_1 \right], \\ \mathcal{L}^{s}_{\text{cyc}} &:= \mathbb{E}_y \left[\mathbb{E}_{\boldsymbol{x}_y} \mathbb{E}_{\boldsymbol{x}_{\bar{y}} \sim \mathbb{P}_{x_{\bar{y}} | h(x_y)}} \left| \boldsymbol{x}_y - g(e_{\bar{y}}(g(e_y(\boldsymbol{x}_y),\, s_{\bar{y}}(h(\boldsymbol{x}_1),\, \boldsymbol{x}_{\bar{y}}))),\, s_y(h(\boldsymbol{x}_y),\, \boldsymbol{x}_y)) \right|_1 \right]. \end{aligned} \tag{2.3.5}$$

[1] Different of d, the embedding function, that we introduced in the previous subsection.

Style-consistency loss [Almahairi et al., 2018, Huang et al., 2018]. Regularizes the translation networks to use the style code.

$$\begin{aligned} \mathcal{L}^{f}_{\text{sty}} &:= \mathbb{E}_y \left[\mathbb{E}_{\boldsymbol{x}_y} \mathbb{E}_{\boldsymbol{z}} \left| f_{\bar{y}}(h(\boldsymbol{x}_y),\, \boldsymbol{z}) - s_{\bar{y}}(h(\boldsymbol{x}_y),\, g(e_y(\boldsymbol{x}_y),\, f_{\bar{y}}(h(\boldsymbol{x}_y),\, \boldsymbol{z}))) \right|_1 \right], \\ \mathcal{L}^{s}_{\text{sty}} &:= \mathbb{E}_y \left[\mathbb{E}_{\boldsymbol{x}_y} \mathbb{E}_{\boldsymbol{x}_{\bar{y}} \sim \mathbb{P}_{x_{\bar{y}}|h(\boldsymbol{x}_y)}} \left| s_{\bar{y}}(h(\boldsymbol{x}_y),\, x_{\bar{y}}) - s_{\bar{y}}(h(\boldsymbol{x}_y),\, g(e_y(\boldsymbol{x}_y),\, s_{\bar{y}}(h(\boldsymbol{x}_y),\, x_{\bar{y}}))) \right|_1 \right]. \end{aligned} \tag{2.3.6}$$

Style diversity loss [Yang et al., 2019, Choi et al., 2017]. Regularizes the translation network to produce diverse samples.

$$\begin{aligned} \mathcal{L}^{f}_{\text{sd}} &:= -\mathbb{E}_y \left[\mathbb{E}_{x_y} \mathbb{E}_{\boldsymbol{z},\boldsymbol{z}'} \left| g(e_y(\boldsymbol{x}_y),\, f_{\bar{y}}(h(\boldsymbol{x}_y),\, \boldsymbol{z})) - g(e_y(\boldsymbol{x}_y),\, f_{\bar{y}}(h(\boldsymbol{x}_y),\, \boldsymbol{z}')) \right|_1 \right], \\ \mathcal{L}^{s}_{\text{sd}} &:= -\mathbb{E}_y [\mathbb{E}_{x_y} \mathbb{E}_{\boldsymbol{x}_{\bar{y}}, \boldsymbol{x}'_{\bar{y}} \sim \mathbb{P}_{x_{\bar{y}}|h(x_y)}} \left| g(e_y(\boldsymbol{x}_y,\, s_{\bar{y}}(h(\boldsymbol{x}_y),\boldsymbol{x}_{\bar{y}}))) - g(e_y(\boldsymbol{x}_y, s_{\bar{y}}(h(\boldsymbol{x}_y),\boldsymbol{x}'_{\bar{y}}))) \right|_1] \end{aligned} \tag{2.3.7}$$

Semantic loss. We introduce the following semantic loss as the cross-entropy between the semantic code of the source samples and that of their corresponding generated samples. We use this loss to regularise the generation to be semantically coherent with the source input.

$$\begin{aligned} \mathcal{L}^{f}_{\text{sem}} &:= -\mathbb{E}_y \left[\mathbb{E}_{\boldsymbol{x}_y, \boldsymbol{z}} [h(\boldsymbol{x}_y) \log(h(g(e_y(\boldsymbol{x}_y),\, f_{\bar{y}}(h(\boldsymbol{x}_y),\, \boldsymbol{z}))))] \right], \\ \mathcal{L}^{s}_{\text{sem}} &:= -\mathbb{E}_y \left[\mathbb{E}_{\boldsymbol{x}_y} \mathbb{E}_{\boldsymbol{x}_{\bar{y}} \sim \mathbb{P}_{x_{\bar{y}}|h(x_y)}} [h(\boldsymbol{x}_y) \log(h(g(e_y(\boldsymbol{x}_y),\, s_{\bar{y}}(h(\boldsymbol{x}_{\boldsymbol{y}}),\, \boldsymbol{x}_{\bar{y}}))))] \right]. \end{aligned} \tag{2.3.8}$$

Finally, we combine all our losses and solve the following optimization.

$$\begin{aligned} \arg \min_{g, e_{\cdot}, f_{\cdot}, s_{\cdot}} \arg \max_{d_{\cdot}} \mathcal{L}^{s}_{\text{adv}} + \mathcal{L}^{f}_{\text{adv}} + \lambda_{\text{sty}}(\mathcal{L}^{s}_{\text{sty}} + \mathcal{L}^{f}_{\text{sty}}) + \lambda_{\text{cyc}}(\mathcal{L}^{s}_{\text{cyc}} + \mathcal{L}^{f}_{\text{cyc}}) + \\ \lambda_{\text{sd}}(\mathcal{L}^{s}_{\text{sd}} + \mathcal{L}^{f}_{\text{sd}}) \; + \lambda_{\text{sem}}(\mathcal{L}^{s}_{\text{sem}} + \mathcal{L}^{f}_{\text{sem}}), \end{aligned} \tag{2.3.9}$$

where $\lambda_{\text{sty}} > 0$, $\lambda_{\text{cyc}} > 0$, $\lambda_{\text{sd}} > 0$ and $\lambda_{\text{sem}} > 0$ are hyper-parameters defined as the weight of each losses.

2.4. Experiments

We compare CatS-UDT with other *unsupervised* domain translation methods and demonstrate that it shows significant improvements on the SPUDT and SHDT problems. We then perform ablation and comparative studies to investigate the cause of the improvements on both setups. We demonstrate SPUDT using the MNIST [LeCun and Cortes, 2010] and SVHN [Netzer et al., 2011] datasets and SHDT using Sketches and Reals samples from the DomainNet dataset [Peng et al., 2019]. We present the datasets in more detail and the baselines in Appendix A.1.1 and Appendix A.1.2 respectively.

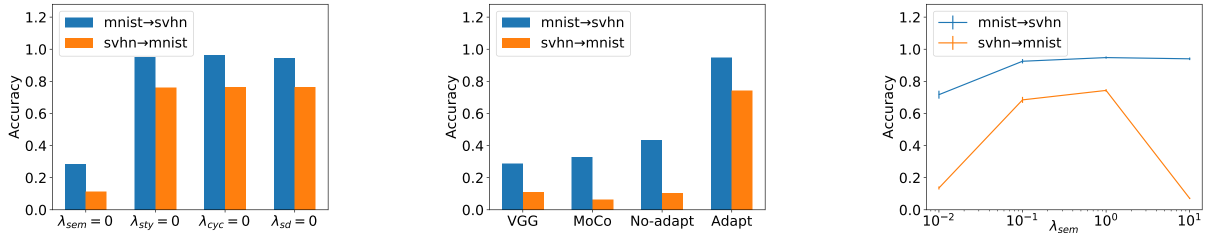
2.4.1. SPUDT with MNIST↔SVHN

Adapted clustering. We first cluster MNIST using IMSAT [Hu et al., 2017]. We reproduce the accuracy of 98.24%. Using the learned clusters as ground-truth labels for MNIST, we adapt the clusters using the VMT [Mao et al., 2019] framework for unsupervised domain adaptation. This trained classifier achieves an accuracy of 98.20% on MNIST and 88.0% on SVHN.

Evaluation. We consider two evaluation metrics for SPUDT. (1) *Domain translation accuracy*, to indicate the proportion of generated samples that have the same semantic category as the source samples. To compute this metric, we first trained classifiers on the

Table 1. Comparison with the baselines. Domain translation accuracy and FID obtained on MNIST (M) ↔SVHN (S) for the different methods considered. The last column is the test classification accuracy of the classifier used to compute the metric. *: Using weak supervision.

	Data	CycleGAN	MUNIT	DRIT	Stargan-V2	EGSC-IT*	CatS-UDT	Target
Acc	M→S	10.89	10.44	13.11	28.26	47.72	**95.63**	98.0
	S→M	11.27	10.12	9.54	11.58	16.92	**76.49**	99.6
FID	M→S	46.3	55.15	127.87	66.54	72.43	**39.72**	-
	S→M	24.8	30.34	20.98	26.27	19.45	**6.60**	-



(a) Ablation study.

(b) Training of semantic encoder.

(c) Varying λ_{sem}.

Fig. 3. Studies on the effect on the translation accuracy on MNIST↔SVHN of (a) Ablating each loss by setting their $\lambda = 0$. (b) Using VGG, MoCO, the presented method for learning categorical semantics without adaptation and with adaptation respectively to train a semantic encoder. (c) Varying λ_{sem}.

target domains. The classifiers obtain an accuracy of 99.6% and 98.0% on the test set of MNIST and SVHN respectively – as reported in the last column of Table 1. (2) FID [Heusel et al., 2017] to evaluate the generation quality.

Comparison with the baselines. In Table 1, we show the test accuracies obtained on the baselines as well as with CatS-UDT. We find that all of the UDT baselines perform poorly, demonstrating the issue of translating samples through a large domain-shift without supervision. However, we do note that StarGAN-V2 obtains slightly higher than chance numbers for MNIST→SVHN. We attribute this to a stronger implicit bias toward identity. EGSC-IT, which uses supervised labels, shows better than chance results on both MNIST→SVHN and SVHN→MNIST, but not better than CatS-UDT.

Ablation study – the effect of the losses In Figure 3a, we evaluate the effect of removing each of the losses, by setting their $\lambda = 0$, on the translation accuracy. We observe that the semantic loss provides the biggest improvement. We run the same analysis for the FID in Appendix A.2.2 and find the same trend. The integration of the semantic loss, therefore, improves the preservation of semantics in domain translation and also improves the generation quality. We also inspect more closely λ_{sem} and evaluate the effect of varying it in Figure 3c. We observe a point of diminishing returns, especially for SVHN→MNIST.

We observe that the reason for this diminishing return is that for a λ_{sem} that is too high, the generated samples resemble a mixture of the source and the target domains, rendering the samples out of the distribution in comparison to the samples used to train the classifier used for the evaluation. We demonstrate this effect and discuss it in more detail in Appendix A.2.2 and show the same diminishing returns for the FID.

Comparative study – the effect of the semantic encoder. In Figure 3b, we evaluate the effect of using a semantic encoder trained using a VGG [Liu and Deng, 2015] on classification, using a ResNet50 on MoCo [He et al., 2020], to cluster MNIST but not adapted to SVHN and to cluster MNIST with adaptation to SVHN. We observe that the use of an adapted semantic network improves the accuracy over its non-adapted counterpart. In Appendix A.2.2 we present the same plot for the FID. We also observe that the FID degrades when using a non-adapted semantic encoder. Overall, this demonstrates the importance of adapting the network inferring the semantics, especially when the domains are sufficiently different.

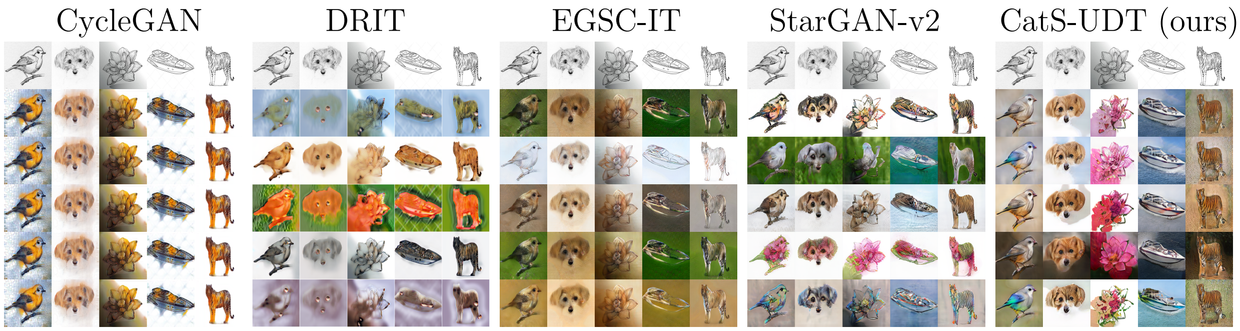


Fig. 4. Comparison with baselines. Comparing the baselines with our approach for translating sketches to real images. For each sketch (top row), we sample 5 different styles generating 5 images in the target domain. For CycleGAN, we copy the generated images 5 times because it is impossible to generate multiple samples in the target domain from the same source image.

Table 2. Comparison with the baselines. Comparing the FID obtained on Sketch→Real for the baselines and our method. We compute the FID per class and over all the categories.

Data	CycleGAN	DRIT	EGSC-IT	StarGAN-V2	CatS-UDT (ours)
Bird	124.10	141.18	101.09	93.58	**92.69**
Dog	170.12	153.05	145.18	108.62	**105.59**
Flower	242.84	223.63	225.24	209.91	**137.01**
Speedboat	189.20	239.94	174.78	127.23	**126.18**
Tiger	156.54	245.73	109.97	69.08	**41.77**
All	102.37	128.45	86.86	65.00	**58.69**

2.4.2. SHDT with Sketches→Reals

Adapted clustering. The representations of the real images were obtained by using MoCo-V2 – a self-supervised model – pre-trained on unlabelled ImageNet. We clustered the learned representation using spectral clustering [Donath and Hoffman, 1973, Luxburg, 2007], yielding 92.13% clustering accuracy on our test set of real images. Using the learned cluster as labels for the real images, we adapted our clustering to the sketches by using a domain adaptation framework – VMT [Mao et al., 2019] – on the representation of the sketches and the reals. This process yields an accuracy of 75.47% on the test set of sketches and 90.32% on the test set of real images. More details are presented in Section 1.1.2.

Evaluation. For the Sketch→Real experiments, we evaluate the quality of the generations by computing the FID over each class individually as well as over all the classes. We do the former because the translation network may generate realistic images that are semantically unrelated to the sketch translated.

Comparison with baselines. We depict the issue with the UDT baselines in Figure 4. For DRIT and StarGAN-V2, the style is independent of the source image. CycleGAN does not have this issue because it does not sample a style. However, the samples are not visually appealing. The images generated with EGSC-IT are dependent on the source, but the style is not realistic for all the source categories. We quantify the difference in sample quality in Table 2 where we present the FIDs.

Ablation study – the effect of the losses. In Table 3a, we evaluate the effect of setting each of removing each of the losses, by setting their $\lambda = 0$, on the FIDs on Sketches→Reals. As in SPUDT, the semantic loss plays an important role. In this case, the semantic loss encourages the network to use the semantic information. This can be visualized in Appendix A.2.3 where we plot the translation. We see that $\lambda_{\text{sem}} = 0$ suffers from the same problem that the baselines suffered, that is that the style is not relevant to the semantic of the source sample.

Comparative study – the effect of the methods to condition semantics. We compare different methods of using semantic information in a translation network, in Table 3b. *None* refers to the case where the semantics is not explicitly used in the translation network, but a semantic loss is still used. This method is commonly used in supervised domain translation methods such as Bousmalis et al. [2017], Hoffman et al. [2018], Tomei et al. [2019]. *Content* refers to the case where we use categorical semantics, inferred using our method, to condition the content representation. Similarly, we also consider the method

Table 3. Studies on the effect of the translation accuracy on Sketches→Reals on (a) Ablating each loss by setting their coefficient $\lambda = 0$. (b) Methods to condition the translation network on the semantics: Not conditioning, conditioning the content representation with categorical semantics, conditioning the content representation with VGG, and conditioning the style with categorical semantics.

(a) Ablation study of the losses

Data	$\lambda_{\text{sem}} = 0$	$\lambda_{\text{sty}} = 0$	$\lambda_{\text{cyc}} = 0$	$\lambda_{\text{SD}} = 0$
Bird	148.32	94.18	108.68	101.97
Dog	131.35	109.50	120.39	106.24
Flower	211.84	124.37	160.97	154.77
Speedboat	185.11	97.52	127.68	99.67
Tiger	153.03	39.24	52.64	41.55
All	69.19	53.43	67.88	58.47

(b) Method to condition on the semantics.

Data	None	Content	Content(VGG)	Style
Bird	101.88	405.29	129.69	92.69
Dog	142.79	343.62	229.18	105.59
Flower	196.70	323.52	220.72	137.01
Speedboat	160.57	280.47	192.38	126.18
Tiger	57.29	212.69	228.84	41.77
All	81.69	275.21	112.10	58.59

used in Ma et al. [2019], in which the semantics comes from a VGG encoder trained with classification. We label this method *Content(VGG)*. For these two methods, we learn a mapping from the semantic representation vector to a feature-map of the same shape as the content representation and then multiply them element-wise – as done in EGSC-IT. *Style* refers the presented method to modulate the style. First, for *None*, the network generates only one style per semantic class. We believe that the reason is that the semantic loss penalizes the network for generating samples that are outside of the semantic class, but the translation network is agnostic of the semantic of the source sample. Second, for *Content*, the network fails to generate sensible samples. The samples are reminiscent of what happens when the content representation is of small spatial dimensionality. This failure does not happen for *Content(VGG)*. Therefore, from the empirical results, we conjecture that the failure case is due to a large discrepancy between the content representation and the categorical representation in addition to a pressure from the semantic loss. The semantic loss forces the network to use the semantic incorporated in the content representation, thereby breaking the spatial structure. This demonstrates that our method allows us to incorporate the semantics category of the source sample without affecting the inductive bias toward the identity, in this setup.

2.5. Conclusion and discussion

We discussed two situations where the current methods for UDT are found to be lacking - Semantic Preserving Unsupervised Domain Translation and Style Heterogeneous Domain Translation. To tackle these issues, we presented a method for learning domain invariant categorical semantics without supervision. We demonstrated that incorporating domain invariant categorical semantics greatly improves the performance of UDT in these two situations. We also proposed to condition the style on the semantics of the source sample and showed that this method is beneficial for generating a style related to the semantic category of the source sample in SHDT, as demonstrated in Sketches→Reals.

While we demonstrated that using domain invariant categorical semantics improves the translation in the SPUDT and SHDT settings, we re-iterate that the quality of the network used to infer the semantics is important. Efforts on robust machine learning and detections of failures are also important in this setup for countering this failure.

Chapter 3

Soft-discretization for self-supervised learning

Note: *This chapter is an on-going project in collaboration with Christos Tsirigotis, Max Schwarzer, Ankit Vani and Aaron Courville.*

3.1. Introduction

Self-supervised learning has demonstrated the capacity to learn useful embeddings of complex and dense data. While this remarkable achievement has been demonstrated on multiple modalities [Chen et al., 2020a, Saeed et al., 2020, You et al.], the datasets used to make the demonstrations are typically sampled from the same distribution as the training set or fine-tuned on the target distribution. Some works started to evaluate the robustness of the models trained with contrastive objective to distribution shifts and their numbers indicate that there are still work that needs to be done in that area Djolonga et al. [2021], Lee et al. [2021].

In this work, we investigate how to make models trained with a self-supervised objective more robust to systematic distribution shifts. We refer to systematic out-of-distribution generalization as the case where the set of observations we have access to during training covers all the possible attributes marginally but where combinations of attributes are not present in the training set. For example, we may want to evaluate the ability of a neural network to identify the shape of a red cone if it has only seen instances of the white and blue cones and other red objects during its training. We consider domain shifts as the case where the test sets is obtained from applying an intervention that is unobserved in the training set [Wang et al., 2021]. For example, the test set could be a corrupted version of the training data [Hendrycks and Dietterich, 2018] or a different rendering [Hendrycks et al., 2021a].

Our observations align with previous works in that self-supervised methods generalize poorly to datasets with systematic shifts. This observation is consistent across several datasets such as dSprites, Shapes3D, and MPI3D, and across two self-supervised objectives, namely noise contrastive estimation [Hjelm et al., 2019b, Chen et al., 2020a] and BYOL [Grill et al., 2020, Chen and He, 2020].

We propose a module to induce soft-discretization during the training of contrastive models to improve their out-of-distribution generalization. The pressure comes from a set of L softmax bottlenecks on independent vectors of size V, with a temperature parameter that controls for the representation's sparsity. This module can be seen as implicitly inducing a set of pseudo-labels for each sample. The inspiration for this module comes from the language emergence community where they argue that a compositional language, that is, a language in which the semantics of a complex utterance is fully determined by the semantics of its constituents and the syntactic rules, can emerge given pressure for expressivity and a bias for compressibility [Kirby et al., 2015]. The contrastive objective can be seen as pressure for expressivity; for instance, through the repulsion term in the noise contrastive estimation objective. Whereas our soft-discretization module can be seen as an architectural bias for compressibility into a set of L soft-discrete tokens.

We demonstrate that such a module improves the in-distribution and out-of-distribution generalization of contrastive models. We present the SDB bottleneck in more detail in Section 3.3. In Section 3.4, we describe the notion of systematic generalization. In particular, we propose a scheme to create systematic splits of the data. Finally, we present our set of results in Section 3.5.

3.2. Background

Note: *We refer to the discussion on contrastive learning in Section 1.1.2 for a background on Noise Contrastive Estimation and BYOL..*

3.2.1. Systematic generalization

Systematic generalization of neural networks is being increasingly studied. [Lake and Baroni, 2018b] study the capacity of recurrent neural networks to generalize to unseen composition of tokens for the task of grounded navigation and demonstrate failure of these models. Montero et al. [2021] and Schott et al. [2021] study the generalization of neural networks to visual representation task on different systematic out-of-distribution settings – including presenting unseen composition of attributes – and note that a wide variety of training setups achieve poor performance. Related to their works, we propose an approach for defining systematic splits and discuss how to generate *harder* and *easier* splits, allowing us to probe the robustness of the different methods to varying difficulties of training.

More broadly, systematic generalization has been studied in other contexts. In particular, it has been studied in symbolic games [Ren et al., 2020a, Denamganaï and Walker, 2020, Lazaridou et al., 2016] where the aim is to learn a compositional representation. Also, in the context of visual question answering, Bahdanau et al. [2019], de Vries et al. [2019]

also proposed a benchmark to evaluate systematicity. Vani et al. [2021] demonstrated improvement on systematic generalization in visual question answering by leveraging advances from the language emergence community. In the realm of natural language, we also note works on evaluating systematicity [Lake and Baroni, 2018a, Ruis et al., 2020]. Loula et al. [2018] demonstrated that recurrent neural networks could systematically learn to recombine functional words in novel contexts where they receive a large number of instances on the pattern to generalize during training. Finally, closer to our work, Dessì et al. [2021] propose to use a sender-receiver setup with discretized a bottleneck at the output of the sender, as done in Ren et al. [2020b].

3.3. Soft-discretization bottleneck

Contrastive learning objectives can generally be decomposed as inducing a pressure for alignment (expressivity) as well as a pressure for uniformity (compressibility) [Wang and Isola, 2020]. Models implementing contrastive learning objectives generally implement a mixture of explicit and implicit pressures toward expressivity and compressibility. For example, the Noise Contrastive Estimation objective used in many models [Chen et al., 2020a, Hjelm et al., 2019b] induces both a pressure to compress the variability coming from the augmentations as well as a pressure for preventing the samples from learning a degenerate solution. BYOL [Grill et al., 2020] induces the expressive pressure implicitly through renormalization and the use of a target network that is updated through a moving average of an online network. More generally, all those methods also have compressive pressure coming from the architectural choices. In general, the compressive pressure induces the structure [Kirby et al., 2015] that is important, we argue, for the generalization that we observe.

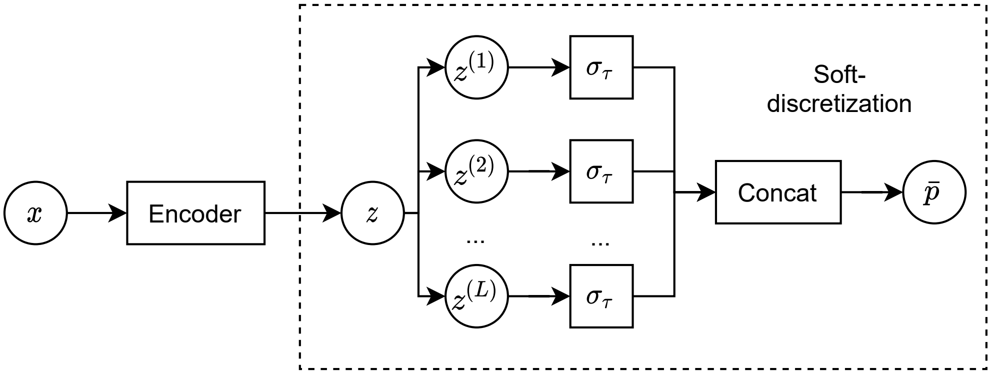


Fig. 1. Proposed Soft-Discretization Bottleneck. σ represents the Softmax operation and s is an linear layer followed by a representation into a $L \times V$ matrix.

Discretization is an efficient way to induce a compressive constraint as the network as a limited bandwidth to encode the data. [Liu et al., 2021] demonstrated that discretization via Vector Quantization improves systematic out-of-distribution generalization on several tasks. However, hard discretization methods such as Vector Quantization [Oord et al., 2018] and REINFORCE [Williams, 1992] generally necessitate the use of straight-through estimation, which typically leads to a harder optimization problem. While hard-discretization has several use-cases, we demonstrate that the generalization benefits can be achieved with a soft variant that permits a smoother optimization.

We illustrate our proposed soft-discretization module in Figure 1. It embeds a representation into a $L \times V$ representation via an embedding module, denoted s. A temperature parameter then scales the logits before re-normalizing via a softmax operation. Concretely, the logits are re-normalized as follows for $i = \{1, \ldots, V\}$:

$$\bar{\boldsymbol{p}}_i^{(k)} := \frac{e^{\boldsymbol{z}_i^{(k)}/\tau}}{\sum_{j=1}^{V} e^{\boldsymbol{z}_j^{(k)}/\tau}}, \tag{3.3.1}$$

for all $k \in \{1, \ldots, L\}$ and τ a temperature parameter.

The soft-discretized bottleneck can be integrated easily in a noise-contrastive estimation model [Hjelm et al., 2019b, Chen et al., 2020a] or BYOL [Grill et al., 2020]. We add it

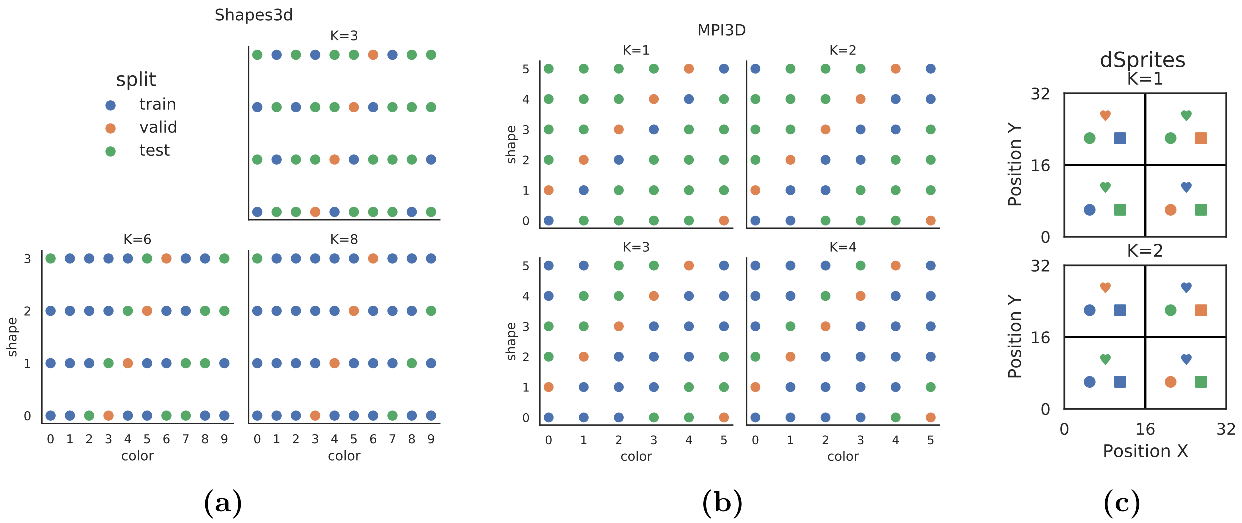


Fig. 2. Systematic splits used by this work with the number of K associated attributes in each split. (a) **Shapes3d** and (b) **MPI3D**: Each shape is associated with K colors in Shapes3d and MPI3D. (c) **dSprites**: Each shape is associated with K quadrant in dSprites.

after the encoder and before the projector in our experiments. The L soft features are concatenated into a vector $\bar{p}$ that is injected into the projector module, which is defined as a Linear layer or a small MLP. Given that small modification, the contrastive objectives are left unchanged.

3.4. Systematic splits

Systematicity, as introduced in the context of understanding the mechanism which enables language and thought [Fodor and Pylyshyn, 1988] in cognitive sciences and the philosophy of the mind, is understood as follows: The ability to understand some natural language sentences is intrinsically connected to the ability to understand certain others. For example, an agent who exhibits systematicity and can understand the concepts of a green triangle and a blue square can understand the concepts of a blue triangle and a green square.

We adapt this definition of systematicity to the task of classification and propose that a model exhibits systematicity if, whenever it recognizes the attributes $\boldsymbol{\pi}_1$ of a sample $\boldsymbol{x}_1$ and the attributes $\boldsymbol{\pi}_2$ of a sample $\boldsymbol{x}_2$ then it can recognize the attributes $\boldsymbol{\pi}'$ of a sample $\boldsymbol{x}'$ where the attributes $\boldsymbol{\pi}'$ is a combinations of attributes in $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$.

Concretely, our definition of systematic splits requires that we have access to the attributes $\{\pi_1, \pi_2, \ldots, \pi_n\}$ of the samples $\boldsymbol{x}$ in a dataset. Assume that each attribute π_i has a set of possible discrete values. Then, we generate a split by associating the value of certain attributes π_i with the value of certain other attributes $\pi_j, j \neq i$, ensuring that all the possible values in π_i for all i are marginally present in each subset. For example, we associate each shape with K different colours in Figure 2a and Figure 2b, while ensuring that each shape and each colour are marginally present in all datasets. In Figure 2c, we associate each shape with K different positional quadrant. All the shapes and possible positions in X and Y are marginally present in the train, valid, and test splits.

While this scheme allows us to study the robustness of models to systematic shifts, it also allows us to control for the strength of the association between two attributes. On one end of the spectrum, an attribute is strongly associated with another attribute if each of its values is associated with one value of the other attribute. This correspond to $K = 1$ in our examples in Figure 2. On the other end of the spectrum, an attribute is weakly associated with another attribute if each of its values is associated with all the possible values of another attribute. This corresponds to where two sets are sampled independently from the same distribution.

3.5. Experiments

We conduct our investigation on in distribution generalization as well as two out-of-distribution generalization categories: systematic out-of-distribution generalization as described in Section 3.4 and generalization to distribution shifts. The comparative experiments on the set of systematic generalization datasets are conducted on the BYOL [Grill et al., 2020] model, the SimCLR model [Chen et al., 2020a] and the SwaV model [Caron et al., 2020] to demonstrate that the proposed module is not contrived to one setup.

The systematic out-of-distribution generalization experiments are performed on a set of three datasets with various systematic splits that we illustrate in Figure 2. We associate two generative factors: shape and colour, for MPI3D [Gondal et al., 2019] and Shapes3D [Burgess and Kim, 2018]. We consider four different splits on MPI3D, ranging from associating one colour with one shape to associating four colours with four shapes. We defined three splits for Shapes3D where we associate each colour with three, six and eight colours. We associate three attributes for dSprites: X position, Y position and shape. The two splits are defined as each shape associated with one or two quadrants. We control for the number of training samples in all the datasets such that the number of training samples in all Ks are the same. That way, a difference in performance between two K is not caused by having more training samples. For Shapes3D, we fixed the wall colour, and the floor colour attributes to a constant value because we observed that not controlling for that variable made the problem of out-of-distribution detection too easy and thus meaningless. We hypothesis that this is due to a transfer of knowledge between the colour attributes between the shape, wall and floor, and thus indicating that these factors are not entirely independent. We leave that exploration for future work.

The distribution shifts experiments are conducted on a set of different distribution shifts on ImageNet. We consider robustness to a set of common image corruptions via Cifar100-C and ImageNet-C [Hendrycks and Dietterich, 2018], natural adversarial examples vias ImageNet-A [Hendrycks et al., 2021b], a set of different rendering of ImageNet's classes via ImageNet-R [Hendrycks et al., 2021a], and a distinct in-distribution set, collected using the same method as the original test set, via ImageNet-v2 [Recht et al., 2019].

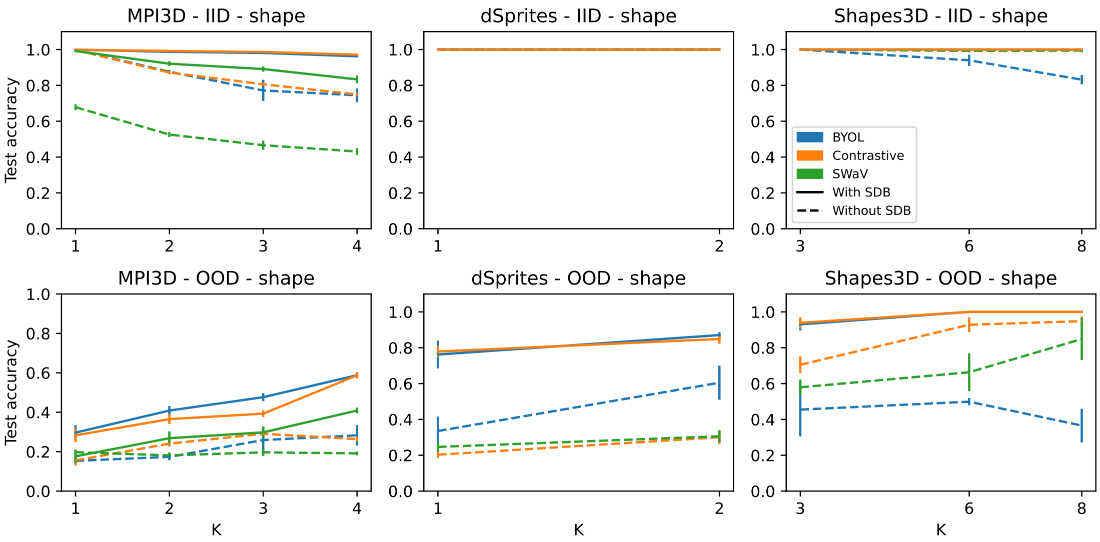


Fig. 3. In-distribution (**Top**) and out-of-distribution (**Bottom**) accuracy on classifying the shape attribute on MPI3D (**Left**), dSprites (**Middle**) and Shapes3D (**Right**) for K associated attributes. K colors for MPI3D and Shapes3D and K positional quadrant for dSprites.

3.5.1. Systematic out-of-distribution generalization

We conduct the systematic out-of-distribution generalization experiments with a simple Convolutional network that we train with ADAM [Kingma and Ba, 2017] and a batch size of 256. For each methods, we perform a hyper-parameters search and pick the best set of parameters according to the systematic validation split for a single K number of associated attributes (see Figure 2). This set of hyper-parameters is used to train every models on every K's number of associated attributes for five seeds. The evaluation is performed by training a linear model with Ridge Regression for each attribute using the training data. This linear model is used to compute the accuracy on in-distribution and out-of-distribution validation and test sets.

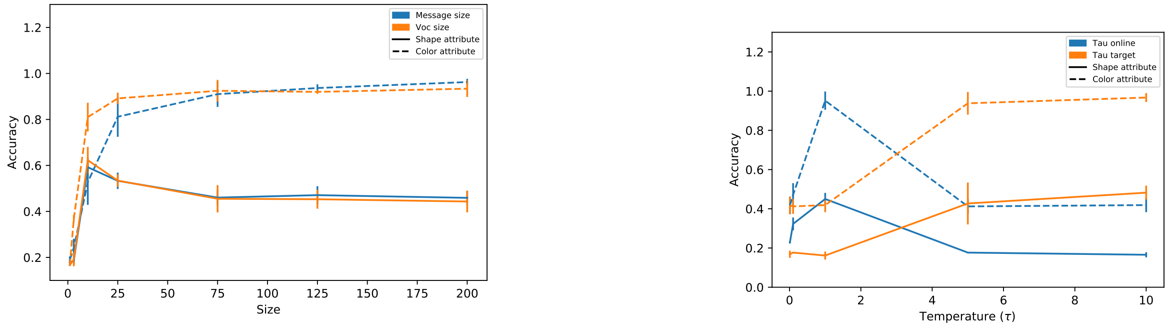
Comparative study. We first present a comparative study across all the subsets presented in this work. We compare the in-distribution and out-of-distribution performance on the shape attribute for the three datasets in Figure 3 and observe a clear pattern. The soft-discretization bottleneck improves the accuracy compared to their baseline for all the methods tested in both in-distribution and out-of-distribution.

Ablation study. We present an ablation study of the module to isolate the factors leading to improved performance. We perform the experiments on the MPI3D dataset with $K = 3$ and present the results in Table 1. We tested the idea of inducing additive and multiplicative noise and noise via dropout to test the hypothesis that a noisy channel might improve the test accuracy. We notice that the noisy channel does not induce improvement in performance in all those cases. Finally, we explore the idea of inducing hard discretization during training and notice that such a procedure substantially reduces performance. The reduction in performance could be due to instability during training, which further motivates the idea of using soft-discretization to induce sparsity.

BYOL	+ noise	Argmax	Softmax	VQ	Test accuracy
✓					0.29 ± 0.01
✓	✓				0.28 ± 0.01
✓			✓		0.52 ± 0.03
✓	✓		✓		0.49 ± 0.04
✓	✓	✓			0.32 ± 0.03
✓				✓	0.39

Table 1. Ablation of SSL-SB on MPI3D-K:3. We test the effect of adding noise, a hard discretization bottleneck via Gumbel-Softmax straight-through estimation and Vector Quantization and the soft discretization bottleneck.

Effect or the message/vocabulary size. We investigate the effect of varying V and L, the vocabulary size and the message size, respectively, on the validation accuracy of predicting the systematically out-of-distribution colour and shape attributes of MPI3D with $K = 3$, in Figure 4a. First, we observe an inverse U-shape curve for the accuracy of the shape attribute with respect to the size of both the vocabulary and the message. This indicates that a bottleneck on the amount of information that the network can pack in the representation leads to an improvement in the out-of-distribution generalization of this attribute. However, we do not see the same behaviour for the colour attribute, indicating that the bottleneck does not lead to better out-of-distribution generalization on this attribute.



(a) Study of the effect of the vocabulary size (V) and the message size (L) on MPI3D for $K = 3$ on the systematically out-of-distribution validation set. We fix the $V = 40$ when interpolating L and $L = 80$ when interpolating V. We evaluate the accuracy on the **shape** and **colour** attributes.

(b) Study of the effect of the temperature parameter on the online (τ_O) and the target (τ_T) networks. We fix the temperature $\tau_O = 1.5$ when interpolating τ_T and $\tau_T = 4.0$ when interpolating τ_O.

Fig. 4. Study of the component of the proposed SDB.

Effect of the temperature on the online and target networks. We also investigate the effect of varying the temperature parameters for both the online and the target network of BYOL on the systematic out-of-distribution generalization on both the colour and the shape attribute of MPI3D with $K = 3$. In this setup, we observe an inverse U-curve for both the shape and the colour attribute when varying the temperature parameter of the online network, demonstrating that the sparsity induced by the temperature of the online network affects the out-of-distribution generalization. However, this effect is not observed when varying the temperature of the target network.

	CIFAR100	CIFAR100-C
BYOL	72.57	30.80
BYOL + SDB	74.27	33.88

Table 2. Test accuracy. Comparison of the test accuracy on CIFAR100 and CIFAR100-C. A linear classifier is trained on top of a trained representation using the CIFAR100 training samples.

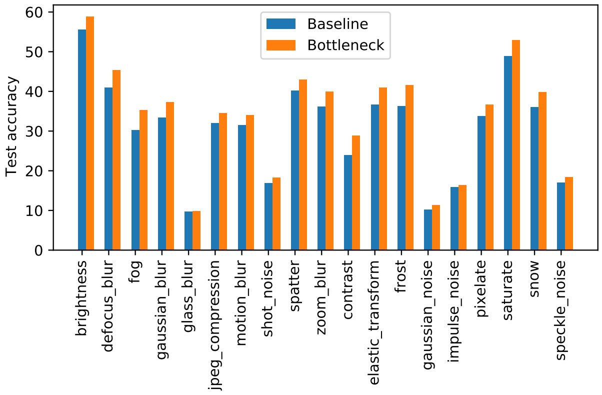


Fig. 5. Fine-grained comparison of the test accuracy on all of the CIFAR100-C domain shift.

3.5.2. Generalization to domain shift

As done in previous works [Grill et al., 2020, Chen and He, 2020, Chen et al., 2020a], we conduct the experiments by training a ResNet-50 [He et al., 2015] using Stochastic Gradient Descent [Bottou et al., 2018] with a cosine decay scheduler on the learning rate. For the ImageNet experiments, we use a batch size of 256 and train for 100 epochs. For each models, we use the other hyper-parameters proposed in their respective paper and tune the hyper-parameters related to our proposed methods using a validation set of 10 samples per classes. Using the hyper-parameters of the model that perform the best on the validation set, we retrain three models with independent random seeds with all the data. The evaluation is done by training a linear classifier, using the training data, on top of the learned representation following the procedure done in previous works.

Comparative study on CIFAR-100. We evaluate the effect of the soft-discretization bottleneck on domain generalization. We train a linear classifier on a pre-trained representation using the in-distribution sample. In Table 2, we present the in-distribution accuracy and the out-of-distribution test accuracy CIFAR100 and CIFAR100-C, respectively. We observe

	Imagenet	Imagenet-v2	Imagenet-r	Imagenet-a	Imagenet-c
BYOL	67.16	53.96	15.35	0.87	33.32
BYOL + SDB	70.22	57.73	17.95	1.01	37.98

Table 3. Test accuracy. Comparison of the test accuracy on ImageNet and and several robustness benchmarks. A linear classifier is trained on top of a trained representation using the ImageNet training samples.

that the bottleneck improves both the in-distribution and out-of-distribution test sets. We present a fine-grained comparison of the accuracy on all of the domain shifts in Figure 5. We observe that the bottleneck induces an improvement across all of the domain shifts.

Comparative study on ImageNet. We perform a similar comparative study on ImageNet using more robustness benchmarks. As demosntrated in Table 3, we observe that the soft-discretization bottleneck improve the in-distribution generalization as well as the robustness for every datasets considered.

3.6. Conclusion

We studied the robustness of models trained with contrastive learning on both systematic out-of-distribution and domain shift. We proposed a drop-in module that that induces soft-sparsity and observe that this module improve the out-of-distribution generalization setups that we studied.

Chapter 4

Conclusion & future works

In this document, we tried to demonstrate how a structured representation of the data can lead to some applications and how it can improve generalization. In the first project, we combined a set of known techniques in the machine learning literature to learn a domain invariant representation of the data. This representation was used in an unsupervised domain translation framework to inform the model of the high-level semantics of the data being transferred and an additional objective to further constraint the optimization to find a solution that preserves the learned semantics. I believe that this kind of approach can be applied more generally to other domains of applications. For example, data in healthcare are notoriously hard to obtain and usually comes from different labs, each of them inducing a distribution shift due, for example, to the use of a different machine. Therefore, learning a representation that is invariant to those domain shifts without explicit supervision could allow practitioners to leverage data from multiple sources.

The language emergence literature heavily inspired the second project presented in this document. One objective of the community studying language emergence is to understand the mechanisms that lead to compositional languages. One direction is that compositional structure in language emerges due to pressures for expressivity and compressivity. Thus, the motivation of this project is to use some of the ideas developed in the language emergence literature and see if these ideas also lead to more structure in representation learning frameworks. The starting idea of this project was to understand how to use Iterated Learning in a contrastive learning framework. While we initially had encouraging results that Iterated Learning yielded improvement in systematic generalization, we realized that the soft-discretization bottleneck that we were applying to bias the model toward having a discrete representation during the learning phase also led to improvement in systematic generalization. Taking a step back, this makes sense since the soft bottleneck induces a compressive bottleneck, which has been demonstrated to lead to structure.

I believe that discrete and soft-discrete bottlenecks in self-supervised learning can open up the opportunity for further improvement in the emergence of structure. We will now explore some future directions that can be fruitful in that direction.

4.1. Directions for further generalization improvement in self-supervised learning

Note 1: *This section contains some preliminary ideas and directions that I believe can be fruitful to extend on the work presented in our ongoing work on self-discrete representation for contrastive learning. Therefore, some propositions are merely speculation backed by intuition, while others may have some empirical evidence to back them up.*

Note 2: *Expected timeline: Exploration of this direction:* ***3 months****.*

Note 3: *Consideration: It might be hard to publish a project without more interesting insights, since these ideas are not novel.*

Iterated learning for contrastive learning. In Section 1.2.1, we have seen an implementation of Iterated Learning in the case of a symbolic communication game. Furthermore, we have seen that Iterated Learning leads to the emergence of compositional structure in the representation generated by the sender. This observation eventually led to the idea presented in chapter 3.

However, before reaching the proposed method, a move naive approach that we attempted was to apply some slight modifications to the Neural Iterated Learning procedure presented in Algorithm 1:

- We define J as a contrastive objective that we minimizes for the encoder,
- The distillation is done by minimizing the cosine similarity between the generated continuous representation of the online network and the representation of the student network,
- When J was defined as the BYOL objective, we would update the parameter of the target network (i.e. the receiver) via a moving average of the parameters of the online network (i.e. the sender).

This led to a continuous version of the Neural Iterated Learning, where the objective was defined as a contrastive objective. However, we noticed that Iterated Learning did not yield improvement over the non-iterated learning, in this case. However, when we induced a hard discrete bottleneck with straight-through estimation, and the distillation was using a cross-entropy between the discrete token of the online network and the prediction of the student network, then Iterated Learning started to yield improvement over the non-iterated learning counterpart. We present an earlier observation of this result in Figure 1 with a set of three different augmentation schemes: random crop, resize and gaussian noise. The experiment was performed on a systematic out-of-distribution of dSprites with $K = 2$, as described in the previous chapter. While some sets of augmentations can lead to an out-of-distribution accuracy, the iterated learning procedure only improves the performance in the discrete case.

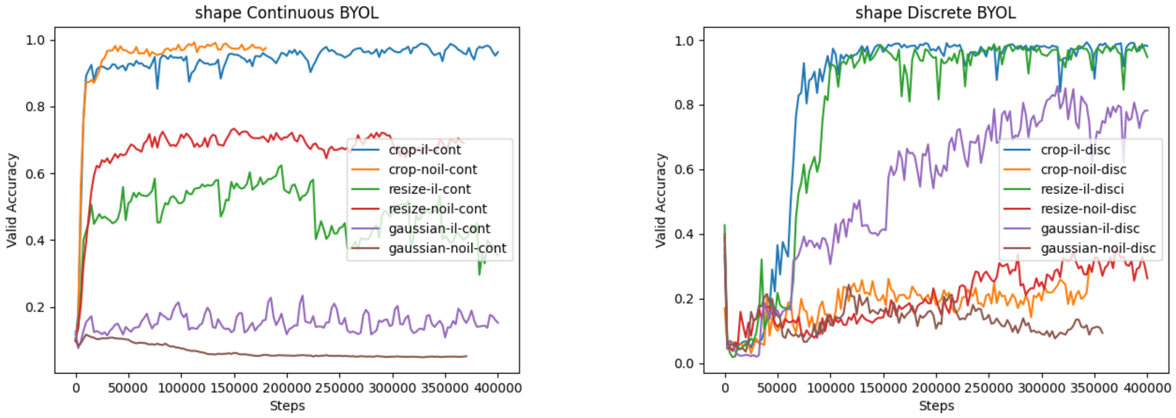


Fig. 1. Comparison on continuous and discrete neural iterated learning using a set on a set of three augmentation scheme.

Therefore, discreteness seems to play an essential role in the success of iterated learning. When we compare with comparable iterated learning methods that exist in the literature, such as Neural Iterated Learning, Noisy Student [Xie et al., 2020] and MILe [Rajeswar et al., 2021], we notice that they all have a discretization component to them. In that sense, this soft-discrete bottleneck could yield better generalization, as demonstrated in these works, but in the context of contrastive learning.

Mixup and cluster assumption for contrastive learning. In Section 1.1.2, we described how the cluster assumption and local lipschitzness improved the task of domain adaptation via several regularizers such as VAT and Virtual Mixup. Mixup and the cluster assumption have also been demonstrated to improve robustness more generally in the context of semi-supervised and domain generalization [Wang et al., 2021]. However, in order to apply these methods, we typically need discrete categories. For example, we cannot enforce the general constraint that the decision boundary has to be in a low-density region if the region is dense. Mixup has not been demonstrated to work well on dense region as far as we can tell. However, contrastive learning methods traditionally learn a dense representation. In that sense, a soft or hard discretization bottleneck would allow us to use known robustness methods to improve the robustness of those methods.

For example, we could implement the Virtual Mixup objective as follow. Take two samples $\boldsymbol{x}$ and $\boldsymbol{x}'$ with $\bar{\boldsymbol{p}}^{L\times V}$ and $\bar{\boldsymbol{p}}'^{L\times V}$ their respective soft labels. Define

$$\tilde{\boldsymbol{x}} := \alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{x}',$$

an interpolated sample with $\alpha \in [0,1]$ and its interpolated soft targets

$$\tilde{\boldsymbol{p}}^{L\times V} := \alpha \bar{\boldsymbol{p}}^{L\times V} + (1-\alpha)\bar{\boldsymbol{p}}'^{L\times V}.$$

Then, we would define the objective as follows

$$\mathcal{L}_{\text{mix}} := -\sum_{i=1}^{L}\sum_{j=1}^{V} \tilde{p}_{i,j}^{\top} \log f(\tilde{x})_{i,j}, \tag{4.1.1}$$

with $f : \boldsymbol{x} \mapsto \bar{\boldsymbol{p}}^{L\times V}$.

4.2. Directions for semantics and structure identification in self-supervised learning

Note 1: *This section contains some speculative ideas and directions that I find interesting and could yield novel ideas.*

Note 2: *Expected timeline: Exploration of this direction: **4 months**.*

Note 3: *Consideration: I don't have much experience in that direction and I might be over-estimating the value or under-estimating its complexity.* Being able to extract the semantics in the data could be valuable. For example, if we could extract the underlying semantics controlling for how the DNA is transcribed into RNA, we could probably understand more about biology. However, I currently believe that if we want to identify semantics or draw relationships between samples via the representation of a learned model using known mathematical tools, then such representation needs to be structured so that the interesting semantic properties are in relation to each other via very simple rules. I also believe that such a structure can emerge given the right set of compressive pressure. After my projects related to language emergence presented in the previous sub-section, understanding how we can identify learned semantics in the representation of a trained network or in the parameters of a Neural Network can be interesting.

At the moment, the preliminary exploration I have performed demonstrates that pruning a trained network via the Lottery Ticket Hypothesis pruning methods unveils the modular structure of the network. In other words, the "winning" ticket is highly modular. To make this observation, I considered this measure of modularity typically used in Network Science and defined it as follows [Clauset et al., 2004]:

$$Q := \sum_{c=1}^{n} \left[\frac{L_c}{m} - \left(\frac{k_c}{2m} \right)^2 \right],$$

where m is the number of edges, c represent each community, and L_c is the number of intra-connection among the member in a community. A community represents as a group of nodes that are tightly connected. I computed the modularity of a trained network given its sparsity and shows that pruning the network unveils a network that is much more modular than its random counterpart, as demonstrated in Figure 2. Interestingly, the capacity to reduce the sparsity is related to the emerging structure in the network.

I believe that a similar observation can be made with representation that generalizes better. The challenge, however, is coming up with both a procedure for identifying the structure in the representation and a heuristic for filtering out unnecessary features. Fortunately, soft

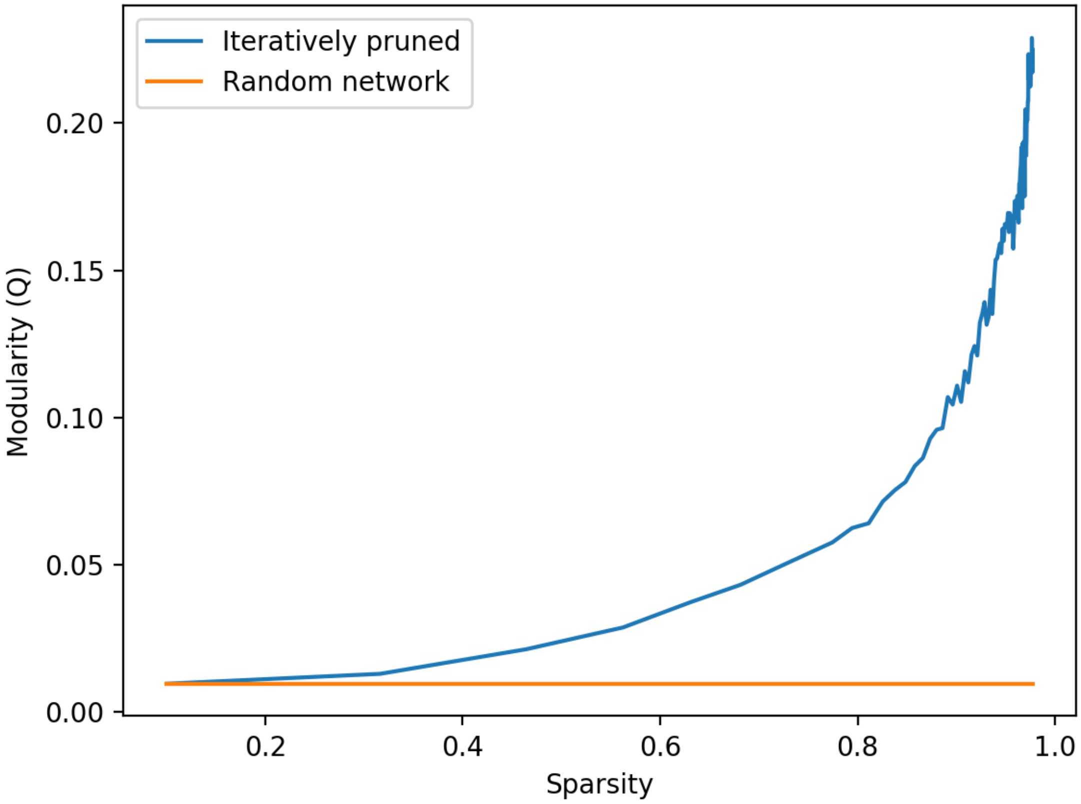


Fig. 2. Comparison of the Modularity of an iteratively pruned network and a random network

and hard discretization might answer the second question. One could say that features with a high entropy capture noise or irrelevant features since higher-order features should rarely be present. More investigation is needed for answering the first question, but one could try using tools from Topological Data Analysis to try and draw structure among the features.

References

Abien Fred Agarap. Deep Learning using Rectified Linear Units (ReLU). *arXiv:1803.08375 [cs, stat]*, February 2019. URL `http://arxiv.org/abs/1803.08375`. arXiv: 1803.08375.

Amjad Almahairi, Sai Rajeshwar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. Augmented CycleGAN: Learning many-to-many mappings from unpaired data. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron C. Courville. Systematic generalization: What is required and can it be learned? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=HkezXnA9YX`.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network Dissection: Quantifying Interpretability of Deep Visual Representations. *arXiv:1704.05796 [cs]*, April 2017. URL `http://arxiv.org/abs/1704.05796`. arXiv: 1704.05796.

Mark A. Bedau. Weak Emergence. *Philosophical Perspectives*, 11:375–399, 1997. doi: 10.1111/0029-4624.31.s11.17.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1), May 2010.

Sagie Benaim, Tomer Galanti, and Lior Wolf. Estimating the success of unsupervised image to image translation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01228-1.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization Methods for Large-Scale Machine Learning. *arXiv:1606.04838 [cs, math, stat]*, February 2018. URL http://arxiv.org/abs/1606.04838. arXiv: 1606.04838.

K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 95–104, 2017.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *arXiv:2104.13478 [cs, stat]*, May 2021. URL http://arxiv.org/abs/2104.13478. arXiv: 2104.13478.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]*, July 2020. URL http://arxiv.org/abs/2005.14165. arXiv: 2005.14165.

Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

Simon Carbonnelle and Christophe De Vleeschouwer. Intraclass clustering: an implicit learning ability that regularizes DNNs. September 2020. URL `https://openreview.net/forum?id=tqOvYpjPax2`.

Gunnar Carlsson and Rickard Brüel Gabrielsson. Topological Approaches to Deep Learning. *arXiv:1811.01122 [cs, math, stat]*, November 2018. URL `http://arxiv.org/abs/1811.01122`. arXiv: 1811.01122.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. 2020.

O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Artificial Intelligence and Statistics*. Max-Planck-Gesellschaft, January 2005.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020b.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020c.

Edward Choi, Angeliki Lazaridou, and Nando de Freitas. Compositional Obverter Communication Learning from Raw Visual Input. February 2018. URL https://openreview.net/forum?id=rknt2Be0-.

Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *CoRR*, abs/1711.09020, 2017.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. *CoRR*, abs/1912.01865, 2019.

Noam Chomsky. A Review of B. F. Skinner's Verbal Behavior, 1959. URL http://cogprints.org/1148/. Issue: 1 Number: 1 Pages: 26-58 Volume: 35.

Morten H. Christiansen and Nick Chater. Language as shaped by the brain. *Behavioral and Brain Sciences*, 31(5):489–509, October 2008. ISSN 1469-1825, 0140-525X. doi: 10.1017/S0140525X08004998. URL https://www.cambridge.org/core/journals/behavioral-and-brain-sciences/article/abs/language-as-shaped-by-the-brain/EA4ABB50915417A1A10569707F574F5E. Publisher: Cambridge University Press.

Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, December 2004. ISSN 1539-3755, 1550-2376. doi: 10.1103/PhysRevE.70.066111. URL http://arxiv.org/abs/cond-mat/0408187. arXiv: cond-mat/0408187.

Abhishek Das, Satwik Kottur, José M. F. Moura, Stefan Lee, and Dhruv Batra. Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning. *arXiv:1703.06585* *[cs]*, March 2017. URL `http://arxiv.org/abs/1703.06585`. arXiv: 1703.06585.

Emmanuel de Bézenac, Ibrahim Ayed, and Patrick Gallinari. Optimal unsupervised domain translation. *arXiv preprint arXiv:1906.01292*, 2019.

Pim de Haan, Taco Cohen, and Max Welling. Natural Graph Networks. *arXiv:2007.08349 [cs, stat]*, November 2020. URL `http://arxiv.org/abs/2007.08349`. arXiv: 2007.08349.

Harm de Vries, Dzmitry Bahdanau, Shikhar Murty, Aaron C. Courville, and Philippe Beaudoin. CLOSURE: assessing systematic generalization of CLEVR models. In *Visually Grounded Interaction and Language (ViGIL), NeurIPS 2019 Workshop, Vancouver, Canada, December 13, 2019*, 2019. URL `https://vigilworkshop.github.io/static/papers/28.pdf`.

Kevin Denamganaï and James Alfred Walker. On (emergent) systematic generalisation and compositionality in visual referential games with straight-through gumbel-softmax estimator. *arXiv preprint arXiv:2012.10776*, 2020.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Roberto Dessì, Eugene Kharitonov, and Marco Baroni. Interpretable agent communication from scratch(with a generic visual processor emerging on the side). *CoRR*, abs/2106.04258, 2021. URL `https://arxiv.org/abs/2106.04258`.

Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, Sylvain Gelly, Neil Houlsby, Xiaohua Zhai, and Mario Lucic. On Robustness and Transferability of Convolutional Neural Networks. *arXiv:2007.08558 [cs]*, March 2021. URL `http://arxiv.org/abs/2007.08558`. arXiv: 2007.08558.

W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973.

Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71, 1988. ISSN 0010-0277.

Tomer Galanti, Lior Wolf, and Sagie Benaim. The role of minimal complexity functions in unsupervised learning of semantic mappings. In *International Conference on Learning Representations*, 2018.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1), 2016.

Martin Gardner. MATHEMATICAL GAMES. *Scientific American*, 223(4):120–123, 1970. ISSN 0036-8733. URL `https://www.jstor.org/stable/24927642`. Publisher: Scientific American, a division of Nature America, Inc.

Ryan Gomes, Andreas Krause, and Pietro Perona. Discriminative clustering by regularized information maximization. In *Neural Information Processing Systems*, 2010.

Muhammad Waleed Gondal, Manuel Wüthrich, DJordje Miladinović, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. 2019.

Abel Gonzalez-Garcia, Joost van de Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1287–1298. Curran Associates, Inc., 2018.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems*, 2014.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Neural Information Processing Systems*. MIT Press, 2005.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf`.

K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. URL `http://arxiv.org/abs/1512.03385`. arXiv: 1512.03385.

Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. September 2018. URL `https://openreview.net/forum?id=HJz6tiCqYm`.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. *arXiv:2006.16241 [cs, stat]*, July 2021a. URL http://arxiv.org/abs/2006.16241. arXiv: 2006.16241.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural Adversarial Examples. pages 15262–15271, 2021b. URL https://openaccess.thecvf.com/content/CVPR2021/html/Hendrycks_Natural_Adversarial_Examples_CVPR_2021_paper.html.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6626–6637. Curran Associates, Inc., 2017.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv:1503.02531 [cs, stat]*, March 2015. URL http://arxiv.org/abs/1503.02531. arXiv: 1503.02531.

Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR 2019*. ICLR, April 2019a.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=Bklr3j0cKX.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In Jennifer G. Dy and Andreas Krause, editors, *ICML*, volume 80, pages 1994–2003. PMLR, 2018.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, January 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90020-8. URL `https://www.sciencedirect.com/science/article/pii/0893608089900208`.

Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *International Conference on Machine Learning*, 2017.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Xun Huang, Ming-Yu Liu, Serge J. Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.

Minyoung Huh, Pulkit Agrawal, and Alexei A. Efros. What makes ImageNet good for transfer learning? *arXiv:1608.08614 [cs]*, December 2016. URL `http://arxiv.org/abs/1608.08614`. arXiv: 1608.08614.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL `https://proceedings.mlr.press/v37/ioffe15.html`.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised Contrastive Learning. *arXiv:2004.11362 [cs, stat]*, March 2021. URL `http://arxiv.org/abs/2004.11362`. arXiv: 2004.11362.

Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, VLDB '04, pages 180–191, Toronto, Canada, August 2004. VLDB Endowment. ISBN 978-0-12-088469-8.

Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. URL `http://arxiv.org/abs/1412.6980`. arXiv: 1412.6980.

Simon Kirby and James R. Hurford. The Emergence of Linguistic Structure: An Overview of the Iterated Learning Model. In Angelo Cangelosi and Domenico Parisi, editors, *Simulating the Evolution of Language*, pages 121–147. Springer, London, 2002. ISBN 978-1-4471-0663-0. doi: 10.1007/978-1-4471-0663-0_6. URL https://doi.org/10.1007/978-1-4471-0663-0_6.

Simon Kirby, Tom Griffiths, and Kenny Smith. Iterated learning and the evolution of language. *Current Opinion in Neurobiology*, 28:108–114, October 2014. doi: 10.1016/j.conb.2014.07.014.

Simon Kirby, Monica Tamariz, Hannah Cornish, and Kenny Smith. Compression and communication in the cultural evolution of linguistic structure. *Cognition*, 141:87–102, August 2015. ISSN 0010-0277. doi: 10.1016/j.cognition.2015.03.016. URL https://www.sciencedirect.com/science/article/pii/S0010027715000815.

Satwik Kottur, José M. F. Moura, Stefan Lee, and Dhruv Batra. Natural Language Does Not Emerge 'Naturally' in Multi-Agent Dialog. *arXiv:1706.08502 [cs]*, August 2017. URL http://arxiv.org/abs/1706.08502. arXiv: 1706.08502.

Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882. PMLR, 10–15 Jul 2018a.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. *arXiv:1711.00350 [cs]*, June 2018b. URL http://arxiv.org/abs/1711.00350. arXiv: 1711.00350.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. *CoRR*, abs/1612.07182, 2016.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of Linguistic Communication from Referential Games with Symbolic and Pixel Input. February 2018. URL `https://openreview.net/forum?id=HJGv1Z-AW`.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database, 2010.

Yann LeCun, Yoshua Bengio, and T Bell Laboratories. Convolutional Networks for Images, Speech, and Time-Series. page 14.

Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Kumar Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation viadisentangled representations. *arXiv preprint arXiv:1905.01270*, 2019.

Kuang-Huei Lee, Anurag Arnab, Sergio Guadarrama, John Canny, and Ian Fischer. Compressive Visual Representations. *arXiv:2109.12909 [cs, math]*, September 2021. URL `http://arxiv.org/abs/2109.12909`. arXiv: 2109.12909.

David K. Lewis. *Convention: A Philosophical Study*. Wiley-Blackwell, 1969.

Dianbo Liu, Alex Lamb, Kenji Kawaguchi, Anirudh Goyal, Chen Sun, Michael Curtis Mozer, and Yoshua Bengio. Discrete-Valued Neural Communication. May 2021. URL `https://openreview.net/forum?id=YSYXmOzlrou`.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 700–708. Curran Associates, Inc., 2017.

S. Liu and W. Deng. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian Conference on Pattern Recognition*

(ACPR), pages 730–734, 2015.

S. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theor.*, 28(2), September 2006.

João Loula, Marco Baroni, and Brenden Lake. Rearranging the familiar: Testing compositional generalization in recurrent networks. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 108–114, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5413. URL `https://aclanthology.org/W18-5413`.

Ulrike Von Luxburg. A tutorial on spectral clustering, 2007.

Liqian Ma, Xu Jia, Stamatios Georgoulis, Tinne Tuytelaars, and Luc Van Gool. Exemplar guided unsupervised image-to-image translation with semantic consistency. In *International Conference on Learning Representations*, 2019.

Xudong Mao, Yun Ma, Zhenguo Yang, Yangbin Chen, and Qing Li. Virtual mixup training for unsupervised domain adaptation. *arXiv preprint arXiv:1905.04215*, 2019.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8), 2018.

Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Instagan: Instance-aware image-to-image translation. In *International Conference on Learning Representations*, 2019.

Milton Llera Montero, Casimir JH Ludwig, Rui Ponte Costa, Gaurav Malhotra, and Jeffrey Bowers. The role of disentanglement in generalisation. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=`

qbH974jKUVy.

Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim. Image to image translation for domain adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4500–4509, 2018.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. The street view house numbers (svhn) dataset, 2011.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom In: An Introduction to Circuits. *Distill*, 5(3):e00024.001, March 2020. ISSN 2476-0757. doi: 10.23915/distill.00024.001. URL https://distill.pub/2020/circuits/zoom-in.

Michael Oliphant and John Batali. Learning and the Emergence of Coordinated Communication, 1997.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. *arXiv:1711.00937 [cs]*, May 2018. URL http://arxiv.org/abs/1711.00937. arXiv: 1711.00937.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dÁlché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International*

Conference on Computer Vision, pages 1406–1415, 2019.

Ori Press, Tomer Galanti, Sagie Benaim, and Lior Wolf. Emerging disentanglement in auto-encoder based unsupervised image content transfer. In *International Conference on Learning Representations*, 2019.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015.

Sai Rajeswar, Pau Rodriguez, Soumye Singhal, David Vazquez, and Aaron Courville. Multi-label Iterated Learning for Image Classification with Label Ambiguity. *arXiv:2111.12172 [cs]*, November 2021. URL `http://arxiv.org/abs/2111.12172`. arXiv: 2111.12172.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet Classifiers Generalize to ImageNet? In *Proceedings of the 36th International Conference on Machine Learning*, pages 5389–5400. PMLR, May 2019. URL `https://proceedings.mlr.press/v97/recht19a.html`. ISSN: 2640-3498.

Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B. Cohen, and Simon Kirby. Compositional languages emerge in a neural iterated learning model. September 2019. URL `https://openreview.net/forum?id=HkePNpVKPB`.

Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B. Cohen, and Simon Kirby. Compositional languages emerge in a neural iterated learning model. In *International Conference on Learning Representations*, 2020a.

Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B. Cohen, and Simon Kirby. Compositional languages emerge in a neural iterated learning model. In *International Conference on Learning Representations*, 2020b. URL `https://openreview.net/forum?id=HkePNpVKPB`.

Pravakar Roy, Nicolai Häni, and Volkan Isler. Semantics-aware image to image translation and domain transfer. *CoRR*, abs/1904.02203, 2019.

Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M. Lake. A benchmark for systematic generalization in grounded language understanding. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/e5a90182cc81e12ab5e72d66e0b46fe3-Abstract.html`.

Aaqib Saeed, David Grangier, and Neil Zeghidour. Contrastive Learning of General-Purpose Audio Representations. *arXiv:2010.10915 [cs, eess]*, October 2020. URL `http://arxiv.org/abs/2010.10915`. arXiv: 2010.10915.

Lukas Schott, Julius von Kügelgen, Frederik Träuble, Peter V. Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello, and Wieland Brendel. Visual representation learning does not generalize strongly within the same domain. abs/2107.08221, 2021.

Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv:1312.6034 [cs]*, April 2014. URL `http://arxiv.org/abs/1312.6034`. arXiv: 1312.6034.

Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

M. Tomei, M. Cornia, L. Baraldi, and R. Cucchiara. Art2real: Unfolding the reality of artworks via semantically-aware image-to-image translation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5842–5852, 2019.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018a.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018b.

Ankit Vani, Max Schwarzer, Yuchen Lu, Eeshan Dhekane, and Aaron Courville. Iterated learning for emergent systematicity in {vqa}. In *International Conference on Learning Representations*, 2021.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to Unseen Domains: A Survey on Domain Generalization. *arXiv:2103.03097 [cs]*, December 2021. URL `http://arxiv.org/abs/2103.03097`. arXiv: 2103.03097.

M. Wang, G. Yang, R. Li, R. Liang, S. Zhang, P. M. Hall, and S. Hu. Example-guided style-consistent image synthesis from semantic labeling. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1495–1504, 2019.

T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018.

Tongzhou Wang and Phillip Isola. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. *arXiv:2005.10242 [cs, stat]*, November 2020. URL `http://arxiv.org/abs/2005.10242`. arXiv: 2005.10242.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

W. Wu, K. Cao, C. Li, C. Qian, and C. C. Loy. Transgaga: Geometry-aware unsupervised image-to-image translation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8004–8013, 2019.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with Noisy Student improves ImageNet classification. *arXiv:1911.04252 [cs, stat]*, June 2020. URL http://arxiv.org/abs/1911.04252. arXiv: 1911.04252.

Dingdong Yang, Seunghoon Hong, Yunseok Jang, Tiangchen Zhao, and Honglak Lee. Diversity-sensitive conditional generative adversarial networks. In *International Conference on Learning Representations*, 2019.

Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding Neural Networks Through Deep Visualization. *arXiv:1506.06579 [cs]*, June 2015. URL http://arxiv.org/abs/1506.06579. arXiv: 1506.06579.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph Contrastive Learning with Augmentations. page 12.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, André Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. The visual task adaptation benchmark. *CoRR*, abs/1910.04867, 2019. URL http://arxiv.org/abs/1910.04867.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv:1611.03530 [cs]*, February

2017. URL `http://arxiv.org/abs/1611.03530`. arXiv: 1611.03530.

P. Zhang, B. Zhang, D. Chen, L. Yuan, and F. Wen. Cross-domain correspondence learning for exemplar-based image translation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5142–5152, 2020.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object Detectors Emerge in Deep Scene CNNs. *arXiv:1412.6856 [cs]*, April 2015. URL `http://arxiv.org/abs/1412.6856`. arXiv: 1412.6856.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017a.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, 2017b.

Appendix A

Appendix section 2

A.1. Additional experimental details

Our results on MNIST↔SVHN and Sketches→Reals datasets were obtained using our Pytorch [Paszke et al., 2019] implementation. We provide the code which contains all the details necessary for reproducing the results as well as scripts that will themselves reproduce the results.

Here, we provide additional experimental and technical details on the methods used. In particular, we present the datasets and the baselines used. We follow with a detailed background on IMSAT [Hu et al., 2017] which is used to learn a clustering on MNIST in our MNIST↔SVHN. Next, we give a background on MoCO [Chen et al., 2020c] which is used to learn a representation on the Reals. Then, we provide a background on Virtual Mixup Training, which is the domain adaptation technique that we use to adapt either the MNIST to SVHN or Reals to Sketches. Finally, we provide a method for evaluating the clusters across multiple domains.

A.1.1. Experimental datasets

Throughout our SPUDT experiments, we transfer between both the **MNIST** [LeCun and Cortes, 2010], which we upsample to 32 × 32 and triple the number of channels, and the **SVHN** [Netzer et al., 2011] datasets. We don't alter the SVHN dataset, i.e. we consider 32 × 32 samples with 3 channels RGB without any data augmentation. But, we consider samples with feature values in the range [-1, 1], as it is usually done in the GAN litterature [Radford et al., 2015], for all of our datasets.

We use a subset of **Sketches** and **Reals** from the DomainNet dataset [Peng et al., 2019] to demonstrate the task of SHDT. We use the following five categories of the DomainNet dataset: *bird, dog, flower, speedboat and tiger*; these 5 are among the categories with most samples in both our domains and possessing distinct styles which are largely non-interchangeable. We resized every image to 256 × 256.

A.1.2. Baselines

For our UDT baselines, we compare with CycleGAN [Zhu et al., 2017b], MUNIT [Huang et al., 2018], DRIT [Lee et al., 2019] and StarGAN-V2 [Choi et al., 2019]. We use these baselines because they are, to our knowledge, the reference models for unsupervised domain translation today. But, none of these baselines use semantics. Also, we are not aware of any UDT method that proposes to use semantics without supervision. Hence, we also consider EGST-IT [Ma et al., 2019] as a baseline although it is weakly supervised by the usage of a pre-trained VGG network. EGSC-It proposes to include the semantics into the translation network by conditioning the content representation. It also considers the usage of exemplar, unconditionally of the source sample.

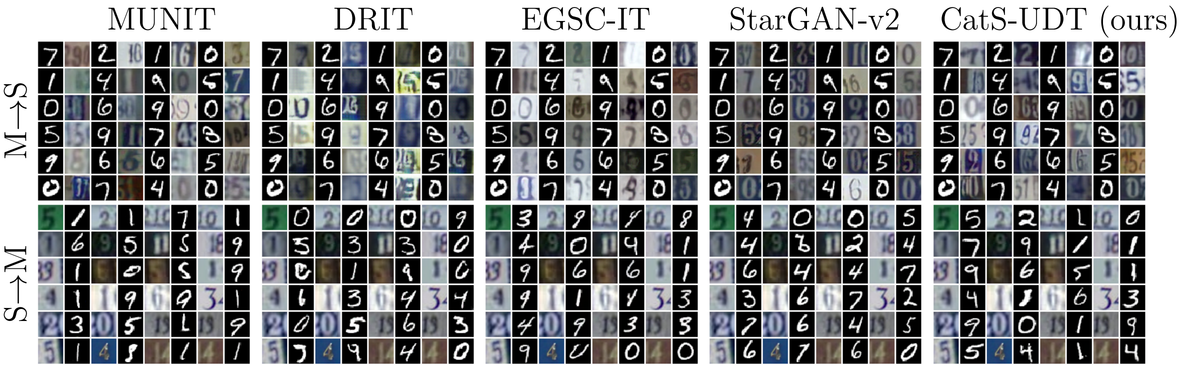


Fig. 1. Qualitative comparison of the baselines with our method on MNIST↔SVHN. Even columns correspond to source samples, and odd columns correspond to their translations.

For each of the baselines, we perform our due diligence to find the set of parameters that perform the best and report our results using these parameters.

A.2. Additional results

A.2.1. Qualitative results for MNIST-SVHN

We present additional qualitative results to provide a better sense of the results that our method achieves. In Figure 1, we show qualitative comparisons with samples of translation for the baselines and our technique. We observe that the use of semantics in the translation visibly helps with preserving the semantic of the source samples. The qualitative results confirm the quantitative results on the preservation of the digit identity presented in Table 1.

Furthermore, in Figure 2, we present qualitative results of the effect of changing the noise sample z on the generation of SVHN samples for the same MNIST source sample. The first row represents the source samples and each column represents a generation with a different z. Each source sample uses the same set of z in the same order. We observe that z indeed grossly controls the style of the generation. Also, we observe that the generations preserve features of the source sample such as the pose. However, we note that some attributes such as typography are not perfectly preserved. In this instance, we conjecture that this is due to the fact the the “MNIST typography" is not the same as the “SVHN typography". For example, the ‘4’s are different in the MNIST and SVHN datasets. Therefore, due to the adversarial loss, the translation has to modify the typography of MNIST.

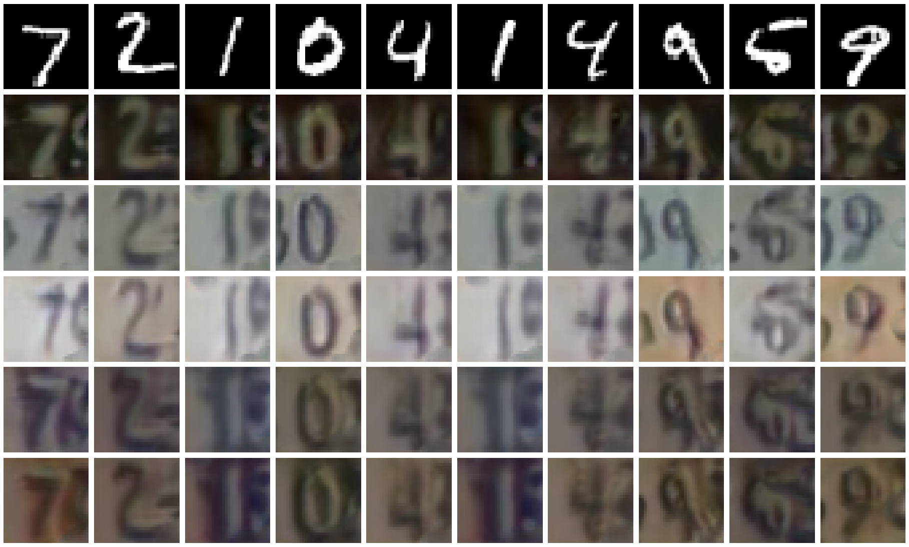
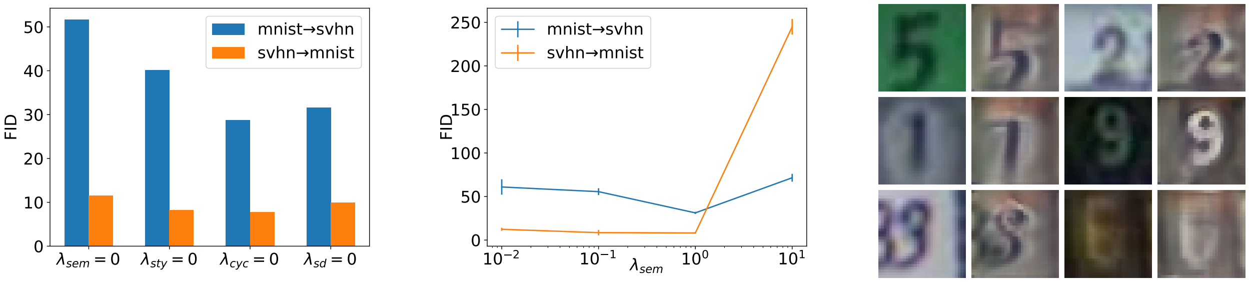


Fig. 2. Multiple sampling for MNIST→SVHN. For each column, the first row is the source sample and each subsequent row is a generation corresponding to a different z.

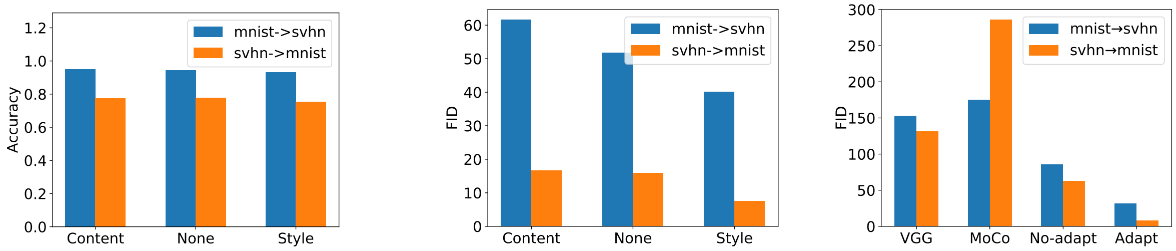


(a) Setting one $\lambda = 0$.

(b) Varying λ_{sem}.

(c) SVHN→MNIST, $\lambda_{\text{sem}} = 10$

Fig. 3. Ablation studies on the effect on the FID on MNIST↔SVHN of (a) Setting one $\lambda = 0$ while keeping the other $\lambda' = 1$, (b) Varying λ_{sem} and (c) Qualitative results of SVHN→ MNIST when $\lambda_{\text{sem}} = 10$.



(a) Accuracy of conditioning methods.

(b) FID of conditioning methods.

(c) FID of semantic encoders.

Fig. 4. Comparative studies on the effect (a) on the translation accuracy and (b, c) on the FID on MNIST↔SVHN on (a, b) Conditioning the content representation on the semantics, not conditioning on semantics and conditioning the style representation on the semantics.

A.2.2. Additional ablation studies for MNIST-SVHN

Ablation study – the effect of the losses on the FID. In Figure 3a, we evaluate the effect of removing each of the losses, by setting their $\lambda = 0$, on the FID. We observe that removing the semantic loss yields the biggest deterioration for the FID. Hence, the semantic loss does not only improve the semantic preservation as observed in Section 2.4.1, but also the image quality of the translation.

Also, we see a U-curve on the FID on MNIST→SVHN with respect to the parameter λ_{sem}. We observe that tuning this parameter allows us to improve the generation quality. We make a similar observation for SVHN→MNIST for both the FID and the accuracy. In Figure 3c, we present qualitative results of the effect of setting $\lambda_{\text{sem}} = 10$. We see that the samples are a mix of MNIST and SVHN samples. The reduction in generation quality explains why we obtain a worst FID when λ_{sem} is too high. Moreover, we see that the generated samples are out-of-distribution, explaining why we obtain a low accuracy although the digit identity is preserved.

Comparative study – effect of the method to condition the semantics. In Figure 4a and in Figure 4b, we evaluate the effect of the method to condition the semantics – in MNIST↔SVHN – on the translation accuracy and on the FID respectively.

None refers to the case where the semantics is not explicitly used to condition any part of the translation network, but the semantic loss is still used. This method is commonly used in supervised domain translation methods such as Bousmalis et al. [2017], Hoffman et al. [2018], Tomei et al. [2019]. *Content* refers to the case where categorical semantics are used to condition the content representation. This method is similar to the method used in Ma et al. [2019], for example, with the exception that the semantic encoder they used is a VGG trained on a classification task. *Style* refers to the case where the categorical semantics are used to condition the style, as we propose to do.

We see that the method to condition the semantics does not affect the translation accuracy on MNIST↔SVHN. However, it does affect the generation quality. This further demonstrates the relevance of injecting the categorical semantics by modulating the style of the generated samples.

Comparative study – effect of adapting the categorical semantics We saw that an adapted categorical semantics improved the semantics preservation on MNST→SVHN in Figure 3b. Here, we will finish the comparison of the effect of adapting the semantics categorical representation on accuracy for SVHN→MNIST and the FID for MNIST↔SVHN in Figure 4c

A.2.3. Additional results for Sketch→Real

We provide more results to support the results presented in Section 2.4.2 on the Sketch→Real task.

Additional quantitative comparisons We observe qualitatively in Table 1 that our method is lacking in terms of diversity with respect to the other methods that do not leverage any kind of semantics. This is not surprising because we penalize the network for generating samples that are unrealistic with respect to the semantics of the source sample.

Effect of setting $\lambda_{\text{sem}} = 0$. We demonstrated that not using the semantic loss considerably degraded the FID, in Table 3a. In Figure 5, we demonstrate qualitatively that the generated samples, when $\lambda_{\text{sem}} = 0$ suffers from the same problem as the baseline: the style is not conditional to the semantics of the source sample.

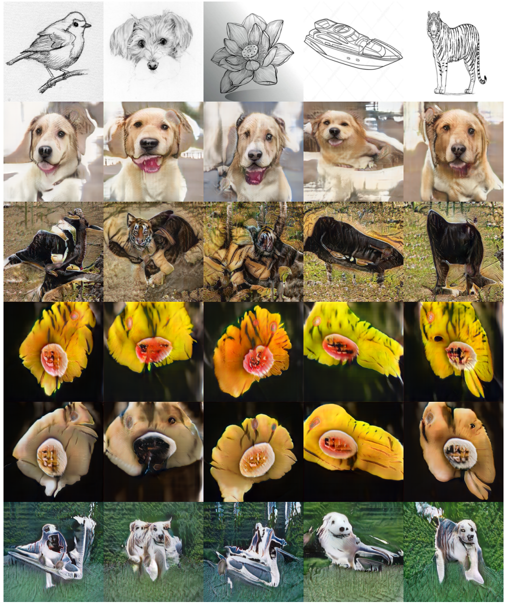


Fig. 5. Sketch→Real using CatS-UDT with $\lambda_{\text{sem}} = 0$. Samples on the first row are the source samples. Samples on the subsequent rows are generated samples.

Table 1. Additional quantitative comparisons with the baselines. We qualitatively compare the baselines using all the Sketches→Reals categories using LPIPS (higher is better), NBD and JSD (lower is better).

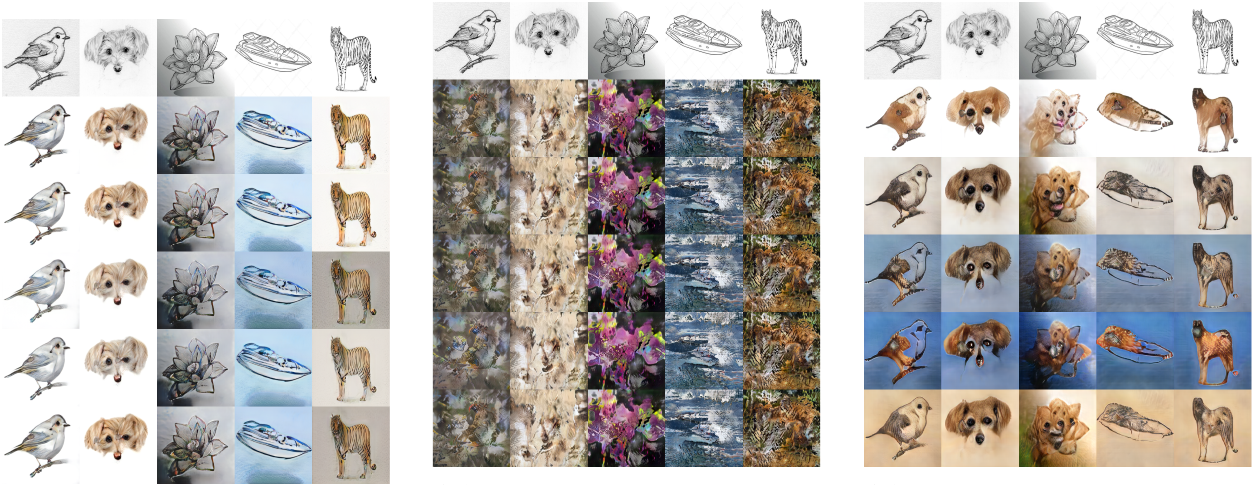
Data	CycleGAN	DRIT	EGSC-IT	StarGAN-V2	CatS-UDT (ours)
LPIPS	0.713	**0.736**	0.064	0.672	0.065
NDB	18	16	19	16	**12**
JSD	0.139	**0.025**	0.044	0.029	0.033

Effect of the method to condition the semantics. The method of conditioning the semantics in the network affects the generation, as observed in Table 3b. We present qualitative results in Figure 6 demonstrating the effect of not conditioning the semantics into any part of the translation network – while still using the semantic loss – and the effect of conditioning the style on the content representation. In the latter case, we consider the semantics as categorical labels adapted to the sketches and the reals as well as semantics defined as the representation from a VGG network trained on classifying ImageNet.

In the first case, the network fails to generate diverse samples and essentially ignores the style input. We conjecture that this happens due to two reasons: (1) The content network and the generator cannot extract the semantics of the source image due to its constraints, relying on the style injected using AdaIN. (2) The mapping network generates the style unconditionally of the source samples; the style for one semantic category might not fit for another (e.g. the style of a tiger does not fit in the context of generating a speedboat). Therefore, to avoid generating, for example, a speedboat with the style of a tiger, the translation network ignores the mapping network.

In the second case, the network fails to generate samples like real images when using categorical semantics. We demonstrate such phenomenon in Figure 6b. The failure is similar to the one observed when the content encoder downsamples the source image beyond a certain spatial dimension. In both these cases, the generated samples lose the spatial coherence of the source image. Without the spatial representation, the generator cannot leverage this information to facilitate the generation. Coupled with the fact that the architecture of the generator assumes access to such a spatial representation and the low number of samples, this explains why it fails at generating sensible samples. In this case, the spatial representation must be lost due to the addition of the categorical semantic representation and the semantic loss. We conjecture that by minimizing the semantic loss, the network tries to leverage the semantic information, interfering with the content representation. Furthermore, we tested a setup similar to the one presented in EGST-IT [Ma et al., 2019] where the semantics is defined as the features of a VGG network in Figure 6c. We see that this failure is not present in this case.

Effect of the spatial dimension of the content representation. We present examples of samples generated when the spatial dimension of the content representation is *too* small to preserve spatial coherence throughout the translation in Figure 7. In this example, we downsample until we reach a spatial representation of 4 × 4 for both our method and CycleGAN. We included CycleGAN to demonstrate that this effect is not a consequence of our method. In both cases, we see that the translation network fails to properly generate the samples as previously observed and discussed. This further highlights the importance of the inductive biases in these models.

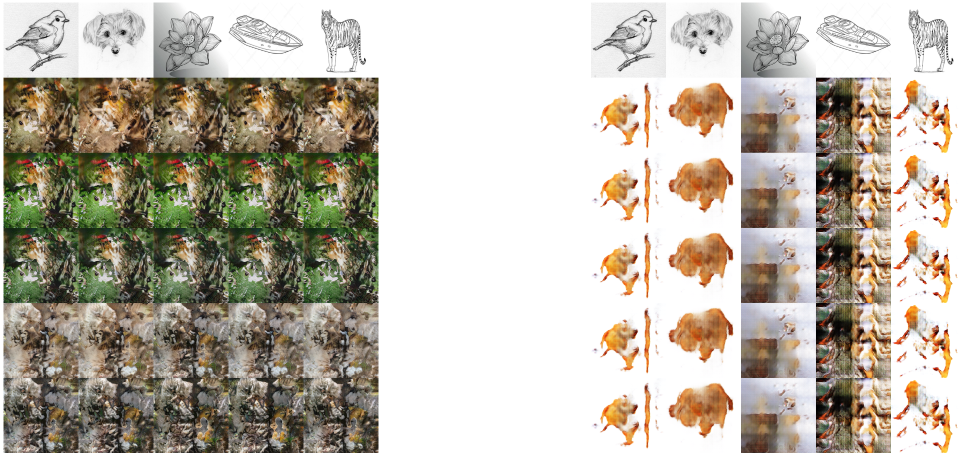


(a) Not conditioning the translation network.

(b) Condition the content representation with categorical semantics.

(c) Condition the content representation with VGG features.

Fig. 6. Qualitative effect of the method to condition the semantics in the translation network in Sketches→Reals. Samples on the first row are the source samples. Samples on the subsequent rows are generated samples.

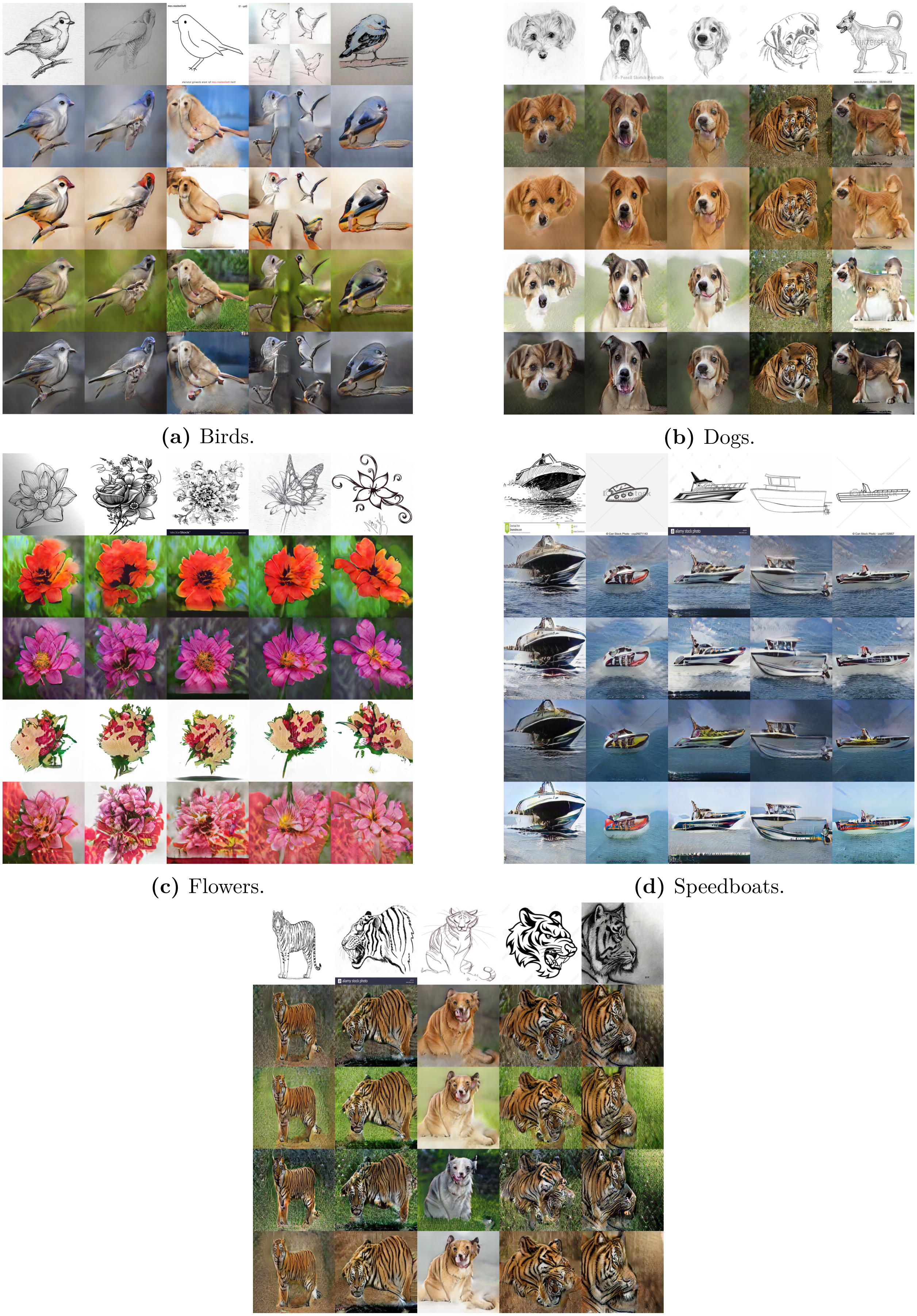


(a) CatS-UDT.

(b) CycleGAN.

Fig. 7. Effect of the representation spatial dimension on the generation of Sketches→Reals. For (a) and (b), we downsample the content representation to a 4×4 feature map. Samples on the first row are the source samples. Samples on the subsequent rows are generated samples.

Additional generation for each classes. We provide additional generations for each of the categories considered in Sketches→Reals in Figure 8 for more test source samples. In the fourth column of the dog panel in Figure 8b and the third column of the tiger panel in Figure 8e, we see a failure case of our method which can happen when a sketch gets mis-clustered. In the first case, the semantic network miscategorizes the dog for a tiger. In the second case, the semantic network miscategorizes the tiger for a dog. This further demonstrates the importance of a semantics network that categorizes the samples with high accuracy for the source and the target domain.



(a) Birds.

(b) Dogs.

(c) Flowers.

(d) Speedboats.

(e) Tigers.

Fig. 8. Additional Sketches→Reals generations for each semantic categories.