2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

CAZSL: Zero-Shot Regression for Pushing Models by Generalizing Through Context

Wenyu Zhang[1], Skyler Seto[1] and Devesh K. Jha[2]

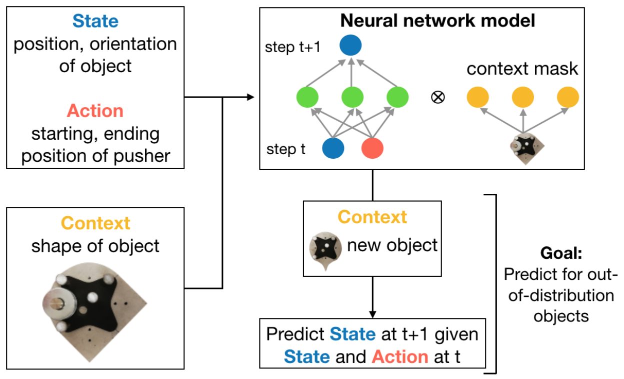
State
position, orientation of object
Action
starting, ending position of pusher
Context
shape of object
Neural network model
step t+1
step t
context mask
⊗
Context
new object
Predict State at t+1 given State and Action at t
Goal:
Predict for out-of-distribution objects

Fig. 1: The proposed idea of learning context-aware zero-shot regression models in the paper. The *context* variables are the additional features which effect the interaction dynamics being considered. The goal is for the learning agent can generalize to different context variables using the proposed approach.

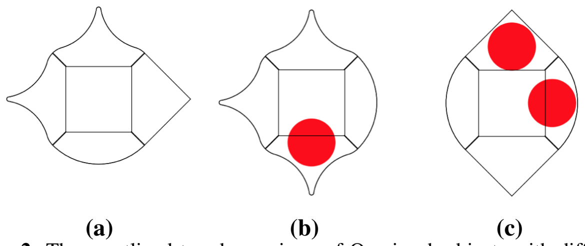

Fig. 2: Three outlined top-down views of Omnipush objects with different shapes and weights. Red circles inside the object indicate the positions of weights. As explained in [2], the pushing dynamics depends on the mass and shape of the object. It is desirable that a learning agent can quickly adapt its notion of pushing interaction based on these attributes of the objects. These attributes of a novel object can be obtained from an auxiliary system (e.g., a vision system). Images are reproduced from `http://web.mit.edu/mcube/omnipush-dataset/` with permission from authors [2].

***Abstract*— Learning accurate models of the physical world is required for a lot of robotic manipulation tasks. However, during manipulation, robots are expected to interact with unknown workpieces so that building predictive models which can generalize over a number of these objects is highly desirable. In this paper, we study the problem of designing deep learning agents which can generalize their models of the physical world by building context-aware learning models. The purpose of these agents is to quickly adapt and/or generalize their notion of physics of interaction in the real world based on certain features about the interacting objects that provide different *contexts* to the predictive models. With this motivation, we present context-aware zero shot learning (CAZSL, pronounced as *casual*) models, an approach utilizing a Siamese network architecture, embedding space masking and regularization based on context variables which allows us to learn a model that can generalize to different parameters or features of the interacting objects. We test our proposed learning algorithm on the recently released *Omnipush* datatset that allows testing of meta-learning capabilities using low-dimensional data. Codes for CAZSL are available at `https://www.merl.com/research/license/CAZSL`.**

I. Introduction

Designing learning agents that can reliably perform robotic manipulation tasks is challenging [1]. One of the reasons among many others is that robotic manipulation deals with a lot of challenging phenomena such as unilateral contacts, frictional contacts, impact, and deformation. These phenomena are challenging to understand or model even when considered individually, and manipulation requires considering several of these simultaneously. Consequently, it is difficult to either derive or learn precise models of interaction that can model different robotic manipulation tasks. Furthermore, robots are expected to interact with unknown workpieces so that building predictive models which can generalize over objects is highly desirable and of practical value [1]. For example, Figure 2 shows objects from the Omnipush dataset [2] where the pushing dynamics depend on the shape and mass distribution of the objects being pushed. While humans generalize effortlessly to variation in different physical properties of objects during interaction, it is difficult for robots to understand this generalization during interaction [3], [4], [5], [6].

Learning accurate models of the physical world is pre-requisite for many model-based robotic manipulation tasks. The motivation of our work is to train general purpose AI agents that can adapt their model of physical systems (e.g., interaction) using some extra features which can be easily obtained using auxiliary systems. For example, the interaction dynamics between two objects can depend on their mass, shape, size, etc. These features for a new object can be estimated using state-of-the-art vision systems or encoded into state-representation features [7]. The learning agents can then adapt their notion of the interaction physics based on these additional inputs. This is very similar to how humans adapt their model of objects based on features that they can sense. Throughout the paper, we call these additional features as *context*. We propose zero-shot regression models outlined pictorially in Figure 1 which are neural networks trained using these additional context variables. The concepts of meta-learning and zero-shot learning are very popular in machine learning literature [8], and are increasingly applied

[1]Wenyu Zhang and Skyler Seto are with Cornell University Department of Statistics and Data Science `{wz258,ss3349}@cornell.edu`
[2]Devesh K. Jha is with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA 02139 `jha@merl.com`

in robotics [7], [9]. For the pushing application, while meta-learning requires support samples from unseen objects for adaptation [2], we do not assume these new samples to be available in our work.

Supervised deep learning models are increasingly popular to model complex relationships in physical systems [10], [11]. The advantage of deep learning models lies in their superior ability to learn complex, non-linear spatial and temporal behaviors through the choice of large network architectures, which can then be optimized using large amounts of data.

However in real applications, we are often unable to collect comprehensive datasets that cover all possible contexts, states and actions. For instance, we may be able to conduct physical experiments with a range of initial conditions for data collection, but not able to observe for all possible initial conditions. Inductive biases typically allow deep learning models to generalize well to further samples collected under the same contexts. This renders such models suitable for applications with a finite and fixed set of contexts. However, they may fare poorly with out-of-distribution samples from unseen contexts due to the lack of ability to generalize across contexts [12], and hence need additional procedures such as context identifiers to correct for this [10].

Inspired by these problems, we present a context-aware zero-shot learning method, CAZSL, for learning predictive models that can generalize across object-dependent context variables. We present a novel combination of context-based mask and regularizer that augments model parameters based on contexts and constrains the zero-shot model to predict similar behavior based on similarity in contexts. This allows us to make accurate prediction on novel objects by adapting the model based on the newly available context. The results of the proposed CAZSL method is reported using the Omnipush dataset which provides a diverse dataset with different objects for pushing dynamics. The proposed idea is presented pictorially in Figure 1, which presents the idea of CAZSL for the Omnipush dataset to generalize over the shape of objects for pushing. We demonstrate empirically that CAZSL improves performance or performs comparably to meta-learning and baselines methods in numerous scenarios.

Contributions: The proposed paper has the following contributions.

1) We present a context-aware zero-shot learning (CAZSL) modeling approach with the motivation of building agents that can quickly adapt their notion of physics based on object-dependent context (analogous to parametric representation). We propose a novel combination of context mask and regularizer to constrain the model using similarities between contexts.
2) We compare the proposed algorithm against several others methods on the recently released Omnipush dataset [2] providing new benchmark results for generalization.

Note that this paper only shows results for modeling using the proposed zero-shot learning approach. Use of the proposed models for model-based control is deferred to a future publication.

II. Related Work

The work presented in this paper is mainly motivated by the goal of creating generalizable models for learning complex interaction dynamics. These kind of physical interactions are common in a lot of physical systems. Interaction between objects especially plays a big role in robotic manipulation where a robot interacts with its environment using selective contacts [1]. Learning accurate predictive models of physical systems and interactions is a very active area of research in robotics and machine learning [13].

Model learning has been studied extensively in both machine learning as well as robotics community. The goal of these techniques in robotic manipulation is to learn high-precision models of interaction of the robot with the physical world which can be then used for synthesizing controllers [14], [15], [16], [17], [18]. Among the possible ways to manipulate an object, pushing stands out as one of the most fundamental. As such, it has gained a lot of attention and thus, has been extensively studied [19], [20], [21], [22]. However, creating reliable models for pushing requires good models of friction, contacts, etc. which still remains poorly modeled in most of the state-of-the-art physics engines. As a result, a lot of data-driven approaches have been proposed based on either learning these interaction models entirely from data or augmented with prior physical knowledge [11], [23], [24]. However, the dynamics of pushing interaction is affected by the physical attributes of the object being pushed (e.g., their shape, mass, size, etc.). As a result, the models learned for a particular object may perform poorly on novel objects [2]. Motivated by this problem and to allow study of generalization to different objects, the Omnipush dataset was released recently [2]. We also draw motivation from this problem and present a technique that can generalize to different objects during a manipulation process and thus, can be used to predict the interaction with different kind of objects. With this goal, we propose a zero-shot regression technique that can generalize using contexts available from different objects. This paper focuses on evaluations through the Omnipush dataset, but we believe that the proposed method is general and can be used to study generalization for other kinds of interactions.

Zero-shot learning algorithms in machine learning are primarily focused on classification problems where either the target classes are rare or expensive to obtain, or the number of target classes is large [25]. These methods assume there is a finite number of classes and may not be easily transferable for use in regression settings. The common technique for zero-shot learning is to make use of auxiliary information or semantic representations, such as object attributes [26], [27], [28] and images [29], to assist learning a model that can generalize to unseen classes. The auxiliary information is usually embedded into a latent space, and regularization has been used to make the embedded representations for each class more separable [30]. We apply a novel regularization where the embedding function is learned according to the distance between contexts, thus maintaining an ordering where more similar contexts are embedded closer together. This allows for a continuous spectrum of contexts instead of a finite number of class prototypes.

III. Background

In this section, we provide some background on the relevant learning approaches that will be referred to in the rest of the paper and allows clarity for readers not familiar with these learning approaches.

A. Siamese Networks

A *Siamese neural network* is a network architecture used for learning a feature space describing the similarity between inputs. It consists of two copies of a network which both take a unique input and compute a distance metric between the two generated feature representations representing the similarity of the two inputs [31]. The parameters of the two copied networks are shared, ensuring that inputs which are similar, according to an application-specific definition, result in a lower distance.

Siamese networks have been used for object tracking, one-shot image classification and image matching [32], [33], [34] and in robotics applications such as robotic surgery and indoor navigation [35], [36].

B. Neural Network Masking

A popular method for training deep neural networks to perform well on a new task is fine-tuning, which takes a pre-trained network and re-trains the weights so that the network performs well on the new task. A major limitation of the fine-tuning approach is that the network may no longer perform well on the original task [37]. Recent works demonstrate an alternative approach where a mask m is learned to update the parameters of the network instead. In particular, by considering a fixed backbone L layer network

$$f(x) = f_L(f_{L-1}(\dots f_1(x)\dots))$$

trained on one dataset, and training an additional set of mask weights $m_1(x), \dots m_L(x)$ on a second dataset, the resulting new network $h(x) = h_L(h_{L-1}(\dots h_1(x)\dots))$ where $h_i(x) = f_i \bigotimes m_i(x)$ as illustrated in Figure 3 has been found to achieve state-of-the-art performance on a second task while maintaining performance on the first [38], [39], [40]. We denote the elementwise product as $\bigotimes$. In general, deep neural networks are often highly over-parametrized [41] and a large number of weights or layers are redundant and can be pruned [42] or fine-tuned for better performance. This pruning is typically done with a binary mask [42], [43], and it is shown that learning a binary mask is sufficient for state-of-the-art accuracy [42], [44], [45] in some cases even without training the weights of the network.

In this work, we utilize the pairwise structure of Siamese neural networks to learn masks to better incorporate object contexts into neural network models, such that existing models can better generalize over object properties. By learning

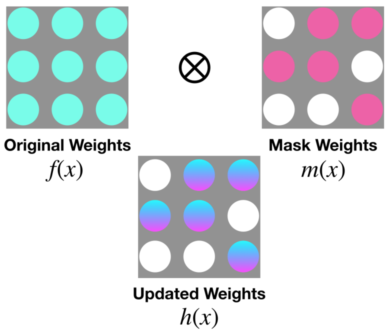

Fig. 3: Overview of masking a deep neural network from [38]. The original weights of the network $f(\cdot)$ are updated by a learned mask $m(\cdot)$ through elementwise product to obtain a new set of weights for the network $h(\cdot)$, allowing the network to specialize for a new set of inputs different from the original.

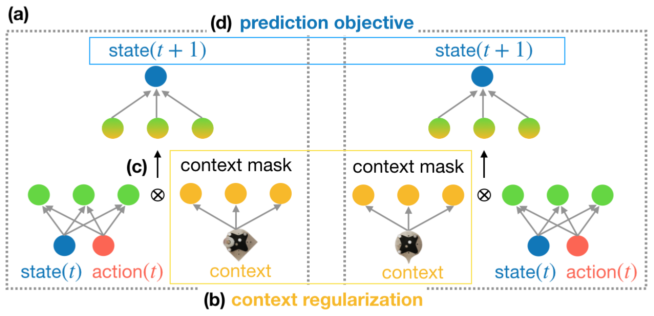

Fig. 4: Proposed CAZSL model; (a) the CAZSL network is a Siamese network which ensures that the same object-push input pairs attain the same predicted state output, (b) regularization on the input context and context embedding mask enforces similar objects to have similar intermediate representations, (c) intermediate representations are altered by the context mask so that the network can guide its output based on knowledge of object properties, (d) the final predicted state is estimated from the masked intermediate representation which incorporates the context.

from pairwise samples of objects, we learn to incorporate the physical properties of objects such that similar objects are in turn pushed similarly because the model parameters are similar.

IV. Proposed CAZSL

In this section, we describe the CAZSL method to effectively incorporate context information into the learning paradigm of neural network models. This allows the learning agents to adapt their model of a real-world physical system based on properties of the interacting objects such as mass and shape, and hence be able to generalize their predictions even towards new unseen objects. The proposed method uses a Siamese network and masking as shown pictorially in Figure 4. Regularization on the context inputs as well as the context mask embedding aims to enforce similar intermediate representations based on similarity in contexts. This idea is explained in more detail in the following text.

A predictive model takes the form of $y_{t+1} = g(x_t; \Theta)$ for a deep neural network model g parameterized by Θ. The inputs at time t are observations x_t. The outputs are denoted y_{t+1}, which are the prediction targets at time $t+1$. However, the Θ learned tends to be biased towards training samples available and the resulting model does not generalize well to out-of-distribution samples. We use different symbols to denote inputs and outputs for generality, although they can contain the same entities.

We learn the model g in an end-to-end fashion to incorporate the ability to generalize to new objects through a novel combination of context mask with a regularization term. We propose learning

$$y_{t+1} = \tilde{g}\left(x_t, c; \tilde{\Theta}\right)$$

where $\tilde{g}(x_t, c) = g_\ell(g_{\ell-1}(\dots g_1(x_t) \bigotimes m(c) \dots))$ is the original ℓ-layered deep neural network with an additional non-linear context mask $m(c)$ which depends on the context c. The context mask is jointly learned by a neural network, and is applied as an elementwise product on the activations from the first layer. The mask augments the embedded input based on context.

Further, we encourage learning $\tilde{\Theta}$ such that if

$$d(c, c') < d(c, c''),$$

then

$$\|m(c) - m(c')\|_F < \|m(c) - m(c'')\|_F$$

for contexts c, c' and c'' where $\|\cdot\|_F$ denotes the Frobenius norm, and d is a suitably chosen distance function defining the magnitude of difference between two contexts. The key idea is that physical dynamics are more similar under more similar contexts. For instance, we would expect the objects in Figure 2a and 2b to behave more similarly to each other when pushed than Figure 2a and 2c, since the first pair of objects shares three common sides while the second pair shares two. The constraint would allow the model to generalize to new out-of-distribution contexts not in the training set, by interpolating or extrapolating based on object attributes observed in the training set. We impose this constraint through the regularization component which we refer to as *context regularization*:

$$\lambda_1 \left(\|m(c) - m(c')\|_F - \lambda_2 d(c, c')\right)^2$$

to be added to the prediction loss in the objective function.

The twin network architecture of Siamese networks allows pairwise comparison of inputs. The network g is trained through a Siamese network structure to optimize the objective function over pairs of inputs $q^{(i)} = \left(x^{(i)}, c^{(i)}\right)$ and $q^{(j)} = \left(x^{(j)}, c^{(j)}\right)$, dropping time indices $t^{(i)}$ and $t^{(j)}$ in the expression for simplicity. The complete loss function for a pair of inputs is:

$$\begin{aligned} &\mathcal{L}\left(q^{(i)}, q^{(j)}; \tilde{\Theta}\right) = \qquad (1) \\ &\frac{1}{2}\left(\tilde{L}\left(q^{(i)}; \tilde{\Theta}\right) + \tilde{L}\left(q^{(j)}; \tilde{\Theta}\right)\right) \\ &\qquad + \lambda_1 \left(\left\|m\left(c^{(i)}; \tilde{\Theta}\right) - m\left(c^{(j)}; \tilde{\Theta}\right)\right\|_F \right. \\ &\qquad \left. - \lambda_2 d\left(c^{(i)}, c^{(j)}\right)\right)^2 \end{aligned}$$

where $\tilde{L}$ is the prediction loss function.

Throughout our experiments, we model

$$p\left(y_{t+1} | q_t, \tilde{\Theta}\right) = \mathcal{N}\left(g\left(x_t, c; \tilde{\Theta}\right); \sigma^2\right)$$

and $\tilde{L}$ is the negative log likelihood of the prediction. Additionally, we consider two distance functions over the vectorized context inputs for our context regularizer:

1) L_2 regularization: Euclidean distance function
$$d\left(c^{(i)}, c^{(j)}\right) = \left\|c^{(i)} - c^{(j)}\right\|_2,$$
2) neural regularization: kernel distance function
$$d\left(c^{(i)}, c^{(j)}\right) = \phi^T\left(c^{(i)}\right)\phi\left(c^{(j)}\right).$$

In the kernel distance function,

$$\phi(x) = W_2 \max(0, W_1 x)$$

is a two layer fully-connected network, or

$$\phi(x) = W \text{avg-pool}\left(\max\left(0, \text{conv}(x)\right)\right)$$

which involves learning the spatial features of x through a convolutional neural network when x is an image. Using L_2 regularization is reasonable when the context variables are continuous, and neural regularization may be more advantageous when the relationship between the context variables are highly non-linear, as in many dynamical systems. Another benefit of the neural regularization is that hyperparameter λ_2 can be absorbed and learned, and we fix $\lambda_2 = 1$ for all experiments with neural regularization which is equivalent to not setting the second hyperparameter.

V. Experiments and Results

We present results to clarify, motivate and justify the use of the proposed CAZSL[1] method for zero-shot learning. To do so, we perform a series of numerical experiments to answer the following questions.

1) Is the inclusion of context helpful towards learning?
2) Does CAZSL improve regression performance on out-of-distribution samples?
3) How should the distance function in CAZSL be selected?

We evaluate our method on a simple regression task as well as six experiments using two contexts from the Omnipush dataset [2]. In the following subsections, we abbreviate competing methods evaluated as ANP (attentive neural process), FCN (fully-connected network), and FCN + CC (FCN with context concatenated to input). ANP is a meta-learning method that uses an attention mechanism on relevant context points for regression [12], and FCN is a 4-layer fully-connected neural network. These two methods are used in the Omnipush data-release paper [2] and they do not make use of context information. We apply our proposed context mask and regularization directly on FCN for easy comparison of their effects. We abbreviate variations of our proposed CAZSL method for ablation studies as FCN + CM

[1]`https://www.merl.com/research/license/CAZSL`

(FCN with context mask), FCN + CM + L2Reg (FCN + CM with L_2 context regularization), and FCN + CM + NeuralReg (FCN + CM with neural context regularization).

We point out that the FCN predicts a Gaussian density for each sample as defined by a mean $\hat{y}_{t+1}$ and standard deviation $\hat{\sigma}$. The mean parameter is evaluated by root-mean-square error (RMSE). We also report the standard deviation (STD) to give a sense of the prediction uncertainty. All values reported correspond to test performance with parameters from the last training epoch.

Hyperparameters: For the simple regression task in Section V-A, we train all models for 500 epochs with the Adam optimizer using a learning rate of $\eta = 0.002$, and a batch size of 32. This configuration is sufficient for convergence due to the small size of the simulation dataset.

For experiments on the Omnipush dataset in Section V-B, we use the same configurations as in [2], that is, we train all models for a maximum of 3000 epochs with the Adam optimizer using a learning rate $\eta = 0.002$, and a batch size of 64. The ANP model in [2] is trained for 5000 epochs with warm-up step of 4000 whereas we train for 3000 epochs with warm-up step of 2000 to match the number of epochs of all other methods. Our replicated results of methods in [2] are comparable with the original results reported.

A. Regression

We use a simple regression task to illustrate the effects of the proposed context mask and regularization on FCN. We simulate univariate Gaussian processes with the RBF kernel:

$$K(a,b) = \xi^2 \exp\left(-\frac{\|a-b\|^2}{2\ell}\right)$$

where ξ controls the scale and ℓ is the bandwidth controlling how far the data can be extrapolated. The parameters are drawn uniformly at random as $\xi \sim \texttt{Unif}(0.1, 10)$ and $\ell \sim \texttt{Unif}(0.1, 10)$. The training set consists a total of 4000 samples extracted from Gaussian processes generated with 200 parameter sets. 20 samples are extracted per parameter set, and denoting z_t as the observation at time t, each sample has predictor $x_t = \{z_{t-2}, z_{t-1}, z_t\}$ which is a subsequence of 3 historical observations and response $y_{t+1} = \{z_{t+1}\}$ as the predictive target. The context variable is the kernel parameters $c = \{\xi, \ell\}$. The test set consists 400 out-of-distribution samples corresponding to 20 new parameter sets.

The simulation is repeated 10 times. We set hyperparameters $\lambda_1 = 0.0001$ and $\lambda_2 = 10$ for the applicable models. We use a small degree of regularization since this regression task is relatively simple and this set of hyperparameters gives good training performance. From Table I, all variations of our proposed method outperform the baselines FCN and FCN + CC on the test set. We note that context concatenation has decreased accuracy while context masking has improved accuracy, which reflects the effectiveness of masking in the embedding space. The use of regularization further improves performance, and FCN + CM + L2Reg has the largest 9.26% reduction of RMSE over FCN.

	RMSE	STD
FCN	0.108	0.131
FCN + CC	0.146	0.133
FCN + CM	0.105	0.121
FCN + CM + L2Reg	**0.098**	0.100
FCN + CM + NeuralReg	<u>0.103</u>	0.113

TABLE I: Average one-step prediction performance on out-of-distribution Gaussian processes samples across 10 simulations. RMSE and STD are based on mean and standard deviation estimates in the Gaussian log-likelihood objective function respectively.

B. Omnipush Dataset

DataSet Description: The Omnipush dataset [2] collected 250 pushes per object for 250 objects on ABS surface (hard plastic). The data collection setup for pushing is shown in Figure 5. The objects are constructed to explore key factors that affect pushing – the shape of the object and its mass distribution – which have not been broadly explored in previous datasets and allow for study of generalization in model learning. Each side of the object has four possible shapes (concave, triangular, circular, rectangular) with three types of extra weights (0g, 60g, 150g). The triangular shape allows two positions (interior, exterior) for extra weights to be attached. A maximum of two weights are attached per object. We denote the shape and mass distribution of the objects as context, and experiment with two types of context variables:

1) Indicator context: length–36 binary vector indicating the shape, extra weight and its position for each side
2) Visual context: numerical array representing top-down view of object displayed in Figure 2.

This allows us to test the generalization capability of our proposed CAZSL technique. The visual context is 32×32, resized from an original 481×481 image[2]. The dataset further has 250 pushes per object for 10 objects on plywood surface. More details of the dataset can be found on the website `http://web.mit.edu/mcube/omnipush-dataset/` and the corresponding paper [2].

The prediction task is to estimate the ending location and orientation of the object after being pushed. In data collection, the pusher is set to move at constant speed. Treating the location and angle of the object as the origin $\left(x_t^{(o)}, y_t^{(o)}, a_t^{(o)}\right) = (0,0,0)$, the model input is $\left(x_t^{(p)}, y_t^{(p)}, a_t^{(p)}\right)$ containing the location and angle of the pusher with respect to the object. As in [2], the location and angle of the object are not needed as input since they are constant at zero with this perspective. The model output is the 3-dimensional vector $\left(x_{t+1}^{(o)}, y_{t+1}^{(o)}, a_{t+1}^{(o)}\right)$. To give a more intuitive representation of model accuracy, we convert RMSE to millimeters by multiplying it by 21.92mm, as done by the authors in [2].

[2]We resize original images in [2] using the resize function with default parameters in scikit-image.

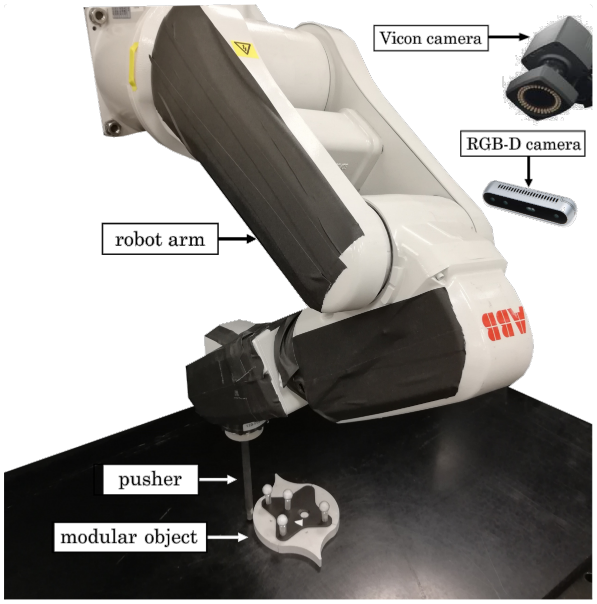

Fig. 5: Omnipush dataset collection setup. The data is collected using an ABB industrial arm. The pusher is a steel rod attached to the end-effector of the arm. This steel arm interacts with the objects which are pushed during the experiments. The picture is reproduced with permission from [2].

Experiment Setups: We use three setups to evaluate generalization performance of models across objects,

1) *Different objects*: training and test objects have different characteristics, that is, the combination of shapes, weights and weight positions for four sides;
2) *Different surfaces*: training objects are pushed on ABS surface and test objects are pushed on plywood;
3) *Different weights*: training and test objects have a different number of extra weights attached.

The *Different surfaces* setup allows evaluation for generalization performance beyond the provided context since surface information is not provided during training. We note that some objects pushed on plywood do not have images provided, and hence we only use indicator context for this setup.

The *Different weights* setup is further split into three sub-setups. There are three possible number of weights $\{0, 1, 2\}$ per object, and we use objects in each of the three options in turn as test objects, and the remaining objects as training objects.

For all experiments with indicator context, we set CAZSL hyperparameters $\lambda_1 = 0.01$ and $\lambda_2 = 10$ where applicable. For visual context, we set $\lambda_1 = 0.01$ and $\lambda_2 = 0.01$ where applicable. The smaller λ_2 is to balance the higher dimensions of the visual context variables. When neural regularization is used, we treat $\lambda_2 = 1$ which is equivalent to not having the second hyperparameter. For context concatenation and context masking, visual contexts are first embedded by a two-layer convolutional neural network learned jointly with the rest of the network.

Results: From Table II and III, CAZSL models consistently have improved performance over the baseline FCN and FCN + CC, reflecting that the inclusion of contextual information helps learning but the context should be applied to the embedding space instead of directly concatenated in the observation space. The RMSE of ANP is consistently between 0.22 and 0.28. Since ANP is a meta-learning method which is aimed at object-generalization, we would expect its performance to be fairly consistent across setups. In comparison with ANP, our CAZSL models achieved better performance except in two sub-cases in Table IIIa for learning *Different weights* with indicator context. However, we note that our L_2 regularized approach outperforms the ANP in the 0 weight test set of the *Different weights* experiment, indicating the ability to better generalize to unknown mass distributions. Moreover, with the use of visual contexts which contain more detailed contextual information, CAZSL models consistently outperform ANP. As expected, visual contexts result in better performance than indicator contexts since the former provides more fine-grained comparison between object shapes.

Comparing between the L_2 and neural regularizations in CAZSL models, we see that the former has lower RMSE when indicator context is used. Since the indicator context is a sparse binary vector, the neural network used for its embedding in neural regularization is possibly over-parameterized, hence resulting in overfitting. When visual context is used, the performance difference between the two choices of regularization is marginal. The convolutional neural network used for context embedding is able to extract spatial features possibly relating to the object geometry and mass distribution, and hence the kernel distance function learned is able to better discern differences between visual contexts. These results suggest that neural regularization might become more suitable with more complex or higher-dimensional context variables.

In summary, in all experiments, we find that our CAZSL models outperform baseline counterparts which do not implement context masking and regularization, and perform comparably or better than the ANP meta-learning baseline. We also observe that using indicator contexts improves performance over using no context most of the time, and using visual contexts improves performance over using indicator contexts or no context in all experiments. This suggests that increasing details in contextual information can be utilized to help learning.

VI. Conclusion and Future Work

Robotic manipulation is hard to model as the interaction dynamics are affected by complex phenomena like dry friction, contacts, impacts, etc. which are difficult to model. Furthermore, the robots are often expected to work with unknown workpieces. As such it is challenging to create models that can predict these interactions accurately over a diverse range of objects with different physical attributes. We present a zero-shot learning method CAZSL which allows us to explicitly consider the physical attributes of different objects so that the predictive model can then be easily adapted to a novel object. We introduced a novel combination of context mask and regularization that augments model

	Different objects					
	Indicator context			Visual context		
	RMSE	STD	Dist. (mm)	RMSE	STD	Dist. (mm)
ANP	0.222	0.079	4.87	0.222	0.079	4.87
FCN	0.330	0.149	7.23	0.330	0.149	7.23
FCN + CC	0.224	0.040	4.91	<u>0.205</u>	0.063	4.49
FCN + CM	<u>0.210</u>	0.043	4.60	**0.193**	0.039	4.23
FCN + CM + L2Reg	**0.205**	0.029	4.49	**0.193**	0.060	4.23
FCN + CM + NeuralReg	0.220	0.037	4.82	**0.193**	0.055	4.23

(a) Different objects: Training and test samples are from objects with different characteristics i.e. shape and mass distribution.

	Different surfaces: Indicator context		
	RMSE	STD	Dist. (mm)
ANP	0.271	0.064	5.94
FCN	0.328	0.154	7.19
FCN + CC	0.264	0.046	5.79
FCN + CM	**0.257**	0.036	5.63
FCN + CM + L2Reg	<u>0.260</u>	0.035	5.70
FCN + CM + NeuralReg	0.263	0.045	5.76

(b) Different surfaces: Training samples are from objects pushed on ABS surface (hard plastic), and test samples are from objects pushed on plywood surface. Some test objects have different characteristics from training objects.

TABLE II: Average method performance under multiple setups where test samples have out-of-distribution properties. RMSE and STD are based on mean and standard deviation estimates in the Gaussian log-likelihood objective function respectively. Setups same as in [2].

	Different weights: Indicator context								
	0 Weight			1 Weight			2 Weights		
	RMSE	STD	Dist. (mm)	RMSE	STD	Dist. (mm)	RMSE	STD	Dist. (mm)
ANP	0.252	0.079	5.52	**0.250**	0.070	5.48	**0.242**	0.079	5.30
FCN	0.327	0.144	7.17	0.356	0.127	7.80	0.329	0.163	7.21
FCN + CC	0.258	0.073	5.66	0.359	0.049	7.87	0.331	0.043	7.26
FCN + CM	<u>0.235</u>	0.038	5.15	0.267	0.033	5.85	<u>0.266</u>	0.030	5.83
FCN + CM + L2Reg	**0.227**	0.040	4.98	<u>0.257</u>	0.034	5.63	0.272	0.033	5.96
FCN + CM + NeuralReg	0.254	0.039	5.57	0.294	0.034	6.44	0.294	0.036	6.44

(a) Different weights: Indicator context.

	Different weights: Visual context								
	0 Weight			1 Weight			2 Weights		
	RMSE	STD	Dist. (mm)	RMSE	STD	Dist. (mm)	RMSE	STD	Dist. (mm)
ANP	0.252	0.079	5.52	0.250	0.070	5.48	0.242	0.079	5.30
FCN	0.327	0.144	7.17	0.356	0.127	7.80	0.329	0.163	7.21
FCN + CC	0.239	0.044	5.15	0.282	0.044	6.18	0.268	0.061	5.87
FCN + CM	<u>0.222</u>	0.034	4.89	0.230	0.032	5.04	<u>0.219</u>	0.039	4.80
FCN + CM + L2Reg	**0.209**	0.079	4.60	**0.220**	0.051	4.82	**0.218**	0.079	4.78
FCN + CM + NeuralReg	**0.209**	0.064	4.56	<u>0.222</u>	0.050	4.87	**0.218**	0.054	4.78

(b) Different weights: Visual context.

TABLE III: Average method performance when training and test samples are from objects with different number of weights attached, out of options from 0 to 2. For example, the heading '0 Weight' means that train samples are from objects with 1 or 2 weights attached, and test samples are from objects have no weight attached. RMSE and STD are based on mean and standard deviation estimates in the Gaussian log-likelihood objective function respectively. Setup not in [2].

parameters based on contexts and constrains the model to predict similar behavior for objects with similar physical attributes. We tested our CAZSL models on the recently released Omnipush dataset. We demonstrate empirically that CAZSL improves performance or performs comparably to meta-learning and object-independent baselines in numerous scenarios.

In the future, we would like to further develop the algorithm and test it on much bigger and diverse interaction datasets. We would like to further investigate the proposed method for multi-step predictive error so that it could be evaluated for control of modeled interactions. Similarly, it would be interesting to test the proposed method for prediction in other physical domains [46], [47].

REFERENCES

[1] M. T. Mason, "Toward robotic manipulation," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 1–28, 2018.

[2] M. Bauza, F. Alet, Y.-C. Lin, T. Lozano-Perez, L. Kaelbling, P. Isola, and A. Rodriguez, "Omnipush: accurate, diverse, real-world dataset of pushing dynamics with rgb-d video," *arXiv*, 10 2019, https://arxiv.org/abs/1910.00618.

[3] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum, "Simulation as an engine of physical scene understanding," *Proceedings of the National Academy of Sciences*, vol. 110, no. 45, pp. 18 327–18 332, 2013.

[4] K. A. Smith, P. W. Battaglia, and E. Vul, "Different Physical Intuitions Exist Between Tasks, Not Domains," *Computational Brain & Behavior*, vol. 1, no. 2, pp. 101–118, 2018.

[5] F. Osiurak and D. Heinke, "Looking for intoolligence: A unified framework for the cognitive study of human tool use and technology," *American Psychologist*, vol. 73, no. 2, pp. 169–185, 2018.

[6] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[7] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *arXiv preprint arXiv:1907.03146*, 2019.

[8] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*. JMLR. org, 2017, pp. 1126–1135.

[9] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *CoRL*, 2020, pp. 1094–1100.

[10] A. Sanchez-Gonzalez, N. M. O. Heess, J. T. Springenberg, J. Merel, M. A. Riedmiller, R. Hadsell, and P. W. Battaglia, "Graph networks as learnable physics engines for inference and control," in *ICML*, 2018.

[11] A. Ajay, J. Wu, N. Fazeli, M. Bauza, L. P. Kaelbling, J. B. Tenenbaum, and A. Rodriguez, "Augmenting physical simulators with stochastic neural networks: Case study of planar pushing and bouncing," in *IROS*. IEEE, 2018, pp. 3066–3073.

[12] H. Kim, A. Mnih, J. Schwarz, M. Garnelo, A. Eslami, D. Rosenbaum, O. Vinyals, and Y. W. Teh, "Attentive neural processes," *arXiv preprint arXiv:1901.05761*, 2019.

[13] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *IJRR*, vol. 32, no. 11, pp. 1238–1274, 2013.

[14] D. Romeres, D. K. Jha, A. DallaLibera, B. Yerazunis, and D. Nikovski, "Semiparametrical gaussian processes learning of forward dynamical models for navigating in a circular maze," in *ICRA*, May 2019, pp. 3195–3202.

[15] A. D. Libera, D. Romeres, D. K. Jha, B. Yerazunis, and D. Nikovski, "Model-based reinforcement learning for physical systems without velocity and acceleration measurements," *arXiv*, 2020, https://arxiv.org/abs/2002.10621.

[16] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *ICRA*. IEEE, 2017, pp. 2786–2793.

[17] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to poke by poking: Experiential learning of intuitive physics," in *NeuRIPS*, 2016, pp. 5074–5082.

[18] N. Fazeli, M. Oller, J. Wu, Z. Wu, J. B. Tenenbaum, and A. Rodriguez, "See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion," *Science Robotics*, vol. 4, no. 26, 2019.

[19] M. T. Mason, "Mechanics and planning of manipulator pushing operations," *IJRR*, vol. 5, no. 3, pp. 53–71, 1986.

[20] K. M. Lynch, H. Maekawa, and K. Tanie, "Manipulation and active sensing by pushing using tactile feedback." in *IROS*, vol. 1, 1992.

[21] K. M. Lynch and M. T. Mason, "Stable pushing: Mechanics, controllability, and planning," *IJRR*, vol. 15, no. 6, pp. 533–556, 1996.

[22] M. A. Peshkin and A. C. Sanderson, "The motion of a pushed, sliding workpiece," *IEEE Journal on Robotics and Automation*, vol. 4, no. 6, pp. 569–598, Dec 1988.

[23] A. Kloss, S. Schaal, and J. Bohg, "Combining learned and analytical models for predicting action effects," *arXiv preprint arXiv:1710.04102*, 2017.

[24] M. Bauza and A. Rodriguez, "A probabilistic data-driven model for planar pushing," in *ICRA*. IEEE, 2017, pp. 3008–3015.

[25] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A survey of zero-shot learning: Settings, methods, and applications," *ACM TIST*, vol. 10, pp. 13:1–13:37, 2019.

[26] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *CVPR*, June 2009, pp. 951–958.

[27] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 3, pp. 453–465, 2013.

[28] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-ucsd birds 200," 2010.

[29] E. Zablocki, P. Bordes, B. Piwowarski, L. Soulier, and P. Gallinari, "Context-Aware Zero-Shot Learning for Object Recognition," in *ICML*, Long Beach, CA, United States, June 2019.

[30] C. Luo, Z. Li, K. Huang, J. Feng, and M. Wang, "Zero-shot learning via attribute regression and class prototype rectification," *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 637–648, Feb 2018.

[31] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *NeuRIPS*, 1994, pp. 737–744.

[32] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *ECCV*. Springer, 2016, pp. 850–865.

[33] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2. Lille, 2015.

[34] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," in *ICPR*. IEEE, 2016, pp. 378–383.

[35] M. Ye, E. Johns, A. Handa, L. Zhang, P. Pratt, and G.-Z. Yang, "Self-supervised siamese learning on stereo image pairs for depth estimation in robotic surgery," *arXiv preprint arXiv:1705.08260*, 2017.

[36] Y. Yeboah, C. Yanguang, W. Wu, and S. He, "Autonomous indoor robot navigation via siamese deep convolutional neural network," in *AIPR*, 2018, pp. 113–119.

[37] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[38] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *ECCV*, 2018.

[39] M. Mancini, E. Ricci, B. Caputo, and S. Rota Bulò, "Adding new tasks to a single network with weight transformations using binary masks," in *ECCV*, 2018, pp. 0–0.

[40] M. Masana, T. Tuytelaars, and J. van de Weijer, "Ternary feature masks: continual learning without any forgetting," *arXiv preprint arXiv:2001.08714*, 2020.

[41] Y. N. Dauphin and Y. Bengio, "Big neural networks waste capacity," *arXiv preprint arXiv:1301.3583*, 2013.

[42] H. Zhou, J. Lan, R. Liu, and J. Yosinski, "Deconstructing lottery tickets: Zeros, signs, and the supermask," in *NeuRIPS*, 2019, pp. 3592–3602.

[43] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," *arXiv preprint arXiv:1803.03635*, 2018.

[44] E. Malach, G. Yehudai, S. Shalev-Shwartz, and O. Shamir, "Proving the lottery ticket hypothesis: Pruning is all you need," *arXiv preprint arXiv:2002.00585*, 2020.

[45] V. Ramanujan, M. Wortsman, A. Kembhavi, A. Farhadi, and M. Rastegari, "What's hidden in a randomly weighted neural network?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 893–11 902.

[46] M. H. Tahersima, K. Kojima, T. Koike-Akino, D. Jha, B. Wang, C. Lin, and K. Parsons, "Deep neural network inverse design of integrated photonic power splitters," *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019.

[47] Y. Zhu and N. Zabaras, "Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification," *Journal of Computational Physics*, vol. 366, pp. 415–447, 2018.