2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Overparametrization helps offline-to-online generalization of closed-loop control from pixels

Mathias Lechner[1], Ramin Hasani[1], Alexander Amini[1], Tsun-Hsuan Wang[1], Thomas A. Henzinger[2], Daniela Rus[1]

***Abstract*— There is an ever-growing zoo of modern neural network models that can efficiently learn end-to-end control from visual observations. These advanced deep models, ranging from convolutional to Vision Transformers, from small to gigantic networks, have been extensively tested on offline image classification tasks. In this paper, we study these vision models with respect to the open-loop training to closed-loop generalization abilities, i.e., deployment realizes a causal feedback loop that is not present during training. This causality gap typically emerges in robotics applications such as autonomous driving, where a network is trained to imitate the control commands of a human. In this setting, two situations arise: 1) Closed-loop testing in-distribution, where the test environment shares properties with those of offline training data. 2) Closed-loop testing under distribution shifts and out-of-distribution. Contrary to recently reported results, we show that *under proper training guidelines*, all vision architectures perform indistinguishably well on in-distribution deployment, resolving the causality gap. In situation 2, We observe that scale is the strongest factor in improving closed-loop generalization regardless of the choice of the model architecture. Our results predict the trend that in the future we will see larger and larger models being used in offline-training-online-deployment imitation learning tasks in robotic applications.**

I. Introduction

End-to-end learning systems are highly suitable in robotics applications because they can efficiently and automatically learn representations from high-dimensional pixel observations without the need for hand-crafting features, bypassing perception to control.

In this space, a tremendous number of advanced deep learning models have been proposed to perform competitively in end-to-end perception-to-control tasks. For example, patch-based vision architectures such as Vision Transformer (ViT) [1] have shown to be competitive with models based on convolutional neural networks (CNNs) [2], [3], [4] in computer vision applications for which CNNs were the predominant choice. A recent line of research, namely the MLPMixer [5], and ConvMixer [6] suggested that the great generalization performance of ViT might be rooted in the patch structure of the inputs rather than the choice of the architecture. There are also works suggesting that self-attention is not crucial in Vision Transformers and simply a gating projection in multi-layer perceptrons (MLPs) [7] or replacing self-attention sublayer with an unparameterized Fourier Transform [8] can outperform ViT.

[1]Massachusetts Institute of Technology (MIT)
mlechner, rhasani, amini, tsunw, rus @mit.edu
[2]Institute of Science and Technology Austria (IST Austria)

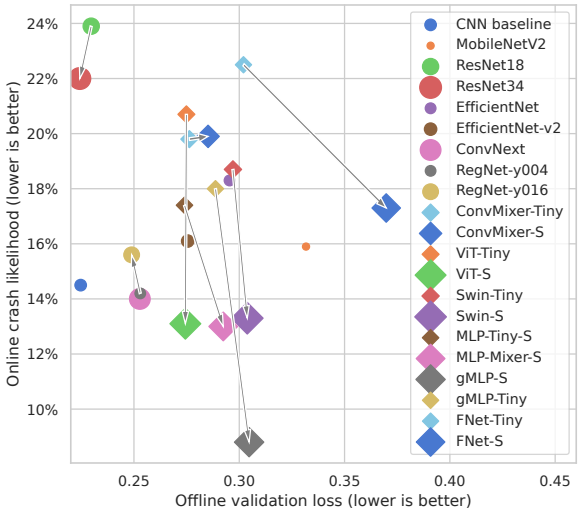

Fig. 1. Online deployment vs. offline training causality gap in perspective. Marker size is linearly proportional to the number of trainable parameters. For models where we tested variants with different sizes, the gray arrows point from small models to their larger counterparts.

These proposals are largely tested in offline settings where the output decisions of the network do not change the subsequent incoming inputs. In other words, patch-based and mixer models trained offline have not yet been evaluated in a closed-loop with an environment where network actions affect next observations, such as in imitation learning (IL) tasks. IL agents typically suffer from a causality gap arising from the transfer of models from open-loop training to closed-loop testing, i.e., introducing a causal relationship between the agent's decision and its future observation, which is not present during training. In this paper, we focus on investigating this gap in a systematic way.

In closed-loop testing, we need to be cognizant of two modes: 1) Closed-loop testing in-distribution. In this setting, we test networks in environments that share similar properties to that of the training environment. 2) Closed-loop testing under distribution shifts and out-of-distribution.

Testing models under both settings requires us to ensure fairness and proper evaluation of the effectiveness of different model architectures in learning robust perception-to-control instances. To this end, we must validate that all baseline models are trained to their best capability given the same amount of hyperparameter optimization budget under a con-

979-8-3503-8457-4/24/$31.00 ©2024 IEEE

trolled training pipeline.

In an attempt to make a unified evaluation of advanced deep learning models in robot imitation learning tasks, in this paper, we set out to design a series of imitation learning pipelines to train models in a controlled and fair setting and test their generalization capability in and out of distribution.

In particular, we design an end-to-end autonomous driving (AD) IL pipeline based on a photorealistic AD simulation platform called VISTA [9], which can test agents in a closed-loop AD environment synthesizing novel views to assess their closed-loop generalization capabilities [10], [11].

Counterintuitively and in contrast to the recently reported results [12], [13], [14], we show that no new architecture is needed to bridge the causality gap between offline training and online testing in-distribution, as our controlled training pipeline enables all models to perform remarkably well on the given tasks. Moreover, for achieving out-of-distribution generalization, we observe that regardless of the model type (Transformer vs CNNs), the causality gap shrinks as we scale up models. These findings confirm the recent theoretical results on the universal law of robustness [15] that as we scale models, they become more robust to perturbations. In our setting, perturbation is analog to the compounding error accumulated at closed-loop testing.

To validate our results and make sure our conclusions are not specific to the AD domain, we further extended our experiments (offline training, then online testing) to more standard visual behavior cloning benchmarks such as perception-to-control arcade learning environment (ALE) [16] tasks. Similar conclusions are then drawn in this case. Our contributions are summarized below:

1) We perform the first systematic study of Vision Transformers architectures and advanced CNNs trained offline using human expert data, followed by a deployment as closed-loop agents in environments with and without modifications of the training data environment.
2) We show that any modern vision architecture can handle the offline training to online generalization gap; however, our results indicate that newer architectures require a more careful selection of their hyperparameters whereas more traditional convolutional architectures tend to work out-of-the-box. Moreover, our results show that no architecture can handle a change in the underlying environment, i.e., a distribution shift, better than any other architecture.
3) We confirm that overparametrization and neural scaling laws observed in standard supervised learning apply to the open-loop to closed-loop generalization as well. Specifically, larger models provide better online generalization at the same offline validation performance as their smaller counterparts, especially in environments with distribution shifts from the training data.

II. Background and Related Works

In this section, we first discuss the image processing architectures studied in this work. Moreover, we recapitulate related works on the understanding of how patch-based CV models process information differently than convolutional architectures. Finally, we discuss existing works on bridging the gap between offline training - online generalization.

Patch-based vision architectures. Motivated by the success of Transformers [17] on natural language processing (NLP) datasets, [1] introduced the *Vision Transformer* (ViT) by adapting the architecture for computer vision tasks. As Transformers operate on a 1-dimensional sequence of vectors, [1] proposed to convert an image into a sequence by tiling it into patches. Each patch is then flattened into a vector by concatenating all pixel values. It has been claimed that vision transforms are much more robust to image perturbations and occlusions [13], [18], as well as be able to handle distribution-shifts [14], [19] better than CNNs. However, more recent works have refuted the robustness claims of Vision Transformers [20] by showing that ViTs can be less robust than convolutional networks when considering carefully crafted adversarial attacks.

Swin Transformer [21] modifies the Vision Transformer by adding a hierarchical structure to the feature sequence of patches. The Swin Transformer applies its attention mechanism not to the full sequence but to a window that is shifted over the entire sequence. By increasing network depth, neighboring windows are merged and pooled into large, less fine-grained windows.

MLP-Mixer [5] adapts the idea of Vision Transformers to map an image to a sequence of patches. This sequence is then processed by alternating plain multi-layer perceptrons (MLP) over the feature and the sequence dimension.

gMLP [7] is a MLP-only vision architecture that differs from the MLP-Mixer by introducing multiplicative spatial gating units between the alternating spatial and feature MLPs.

FNet [8] replaces the learnable spatial mixing MLP of the MLP-Mixer architecture by a fixed mixing step. In particular, a parameter-free 2-dimensional Fourier transform is applied over the sequence and features dimensions of the input.

ConvMixer [6] replace the MLPs of the MLP-mixer architecture by alternating depth-wise and point-wise 1D convolutions. While an MLP mixes all entries of the spatial and feature dimension, the convolutions of the ConvMixer mix only local information, e.g., kernel size was set to 9 in [6].

Advanced convolutional architectures. Here, we briefly discuss modern variants of CNN architectures.

ResNets [22] add skip connections that bypass the convolutional layers. This simple modification allows training much deeper networks than a pure sequential composition of layers. Consequently, skip connections can be found in most modern neural network architecture, including patch-based and advanced convolutional models.

MobileNetV2 [23] replace the standard convolution operations by depth-wise separable convolutions that process the spatial and channel dimension separately. The resulting network requires fewer floating-point operations to compute, which is beneficial for mobile and embedded applications.

EfficientNet [24] is an efficient convolutional neural network architecture derived from an automated neural architecture search. The objective of the search is to find a network

Fig. 2. Visualization of sample observations used in our end-to-end AD experiment, spanning across various seasons and times of the day.

topology that achieves high performance while simultaneously running efficiently on CPU devices.

EfficientNet-v2 [25] fixes the issue of EfficientNets that despite their efficiency on CPU inference, they can be slower than existing architecture types on GPUs at training and inference.

RegNet [26] is a neural network family that systematically explores the design space of previously proposed advances in neural network design. The RegNet-Y subfamily specifically scales the width of the network linearly with depth and comprises squeeze-and-excitation blocks.

ConvNext [27] is a network that subsumes many recent advances in the design of vision architectures, including better activation functions, replacing batch-norm by layer-normalization, and a larger kernel size into standard ResNets.

Imitation learning (IL). IL describes learning an agent from expert demonstrations. The training data consist of observation-action pairs [28] and the learning happens either directly via behavior cloning [29] or indirectly via inverse reinforcement learning [30]. When IL agents are deployed online, they most often deviate from the expert demonstrations leading to compounding errors and incorrect inference. Numerous works have tried to address this problem by adding augmentation techniques that collect data from the cloned model in closed-loop settings. This includes methods such as DAgger [31], [32], state-aware imitation [33], [34], [35], pre-trained policies through meta-learning [36], [37], min-max optimization schemes [29], [38], [39], [40], using insights from causal inference [41], [42], and using a world-model in the loop [43], [44], [45], [46].

OOD generalization. It is fundamentally challenging for statistical models to tackle OOD problems [47], [48], [49], such as domain adaptation [50], [51], [52], [53], [54], debiasing [55], [56], [57], [58], [59], and even practically more challenging settings where OOD semantics are unlabeled [60], [61], [62], [63]. A large body of recently proposed solutions to OOD generalization, explored causal inference such as causal interventions [62], [41], designing counterfactual schemes [64], [65], and using attention-based models [66], [67], [68], [69], [42], [70]. Here, our study aims to explore how advanced vision networks compare in terms of OOD generalization in online closed-loop with their environments, when trained offline.

III. Methodology

In this section, we first describe our recipe for how to systematically train end-to-end imitation learning agents offline via a fair hyperparameter tuning pipeline. We then narrate our experimental setup, followed by the method we use for systematic online testing in and out of distribution.

A. Fair Training Setup

End-to-end deep learning models are typically benchmarked against each other, where one model showed to be outperforming the other. But is it truly the case? Here, we set out to design a controlled offline training to an online testing setup to fairly investigate how advanced vision baselines compare with each other. The training recipe is as follows:

1) We conduct a systematic hyperparameter tuning process (described in detail in the next subsection) for each of the 21 tested advanced deep models individually. In particular, we ran a grid search over the two most influential hyperparameters, the learning rate and the weight decay rate.
2) We do not perform any early stopping but train a substantial number of optimization steps, which has been shown to be vital for generalization, especially on smaller datasets [71], [72], [73], [74].
3) We deploy a custom staircase learning rate decay schedule that decreases the learning rate over the training process by dividing the learning rate by four at 60%, 80%, and 93% of the training epochs.
4) We warm up the training by running the first epochs with 1/10th of the initial learning rate in order to have the moments' estimates in Adam [75], Batch-Normalization [76], and Layer-Normalization [77] modules initialized properly.
5) We replace the standard Adam optimizer with AdamW [78], which decouples the weight decay rate from the loss function, thus avoiding biasing the moments' estimators of Adam.
6) We apply a rich set of data augmentation techniques, including random brightness, contrast, and saturation modifications, guided policy learning [79], and add noise to the expert's actions during exploration [80].

Baselines We compare all neural network architectures introduced in the related work section. Overall, we test 21 advanced models, including nine modern convolutional networks and 12 modern patch-based architectures. The exact architecture of these models is directly taken from their reported research article. Additionally, we included a controlled vanilla CNN baseline that comprises seven convolutional layers, each followed by a batch-normalization layer and a ReLU activation function. The first convolution applies a 5-by-5 kernel with 64 filters. The following convolution layers all apply a 3-by-3 kernel with 128, 128,

256, 256, 512, and 512 filters, respectively. Before the first 256-filters and 512-filters convolutional layers max-pooling layers with stride 2 downsample the feature maps. A global average pooling layer is applied to feature maps of the final convolution layer, followed by a fully-connected layer with 512 units and a dropout layer. There are no skip connections present in the baseline model.

B. Hyperparameter tuning

In order to have a fair comparison, we perform a systematic hyperparameter tuning process for each architecture. Particularly, we run a grid search over the learning rate and regularization factors (weight decay and dropout rate), which have been shown to have the strongest impact on the performance of the neural networks [81], [82], [78]. The objective function of the tuning process is set to the validation loss of the end-to-end driving task. The grid search first searches for the optimal learning rate by evaluating the network with a learning rate of $\{0.01, 0.003, 0.001, 0.0003\}$. Next, the learning rate is fixed to the best performing one, and the search aims to find the right strength of the regularization factor. We evaluate four levels of regularization strengths measured by a pair (w, d), where w is the weight decay factor, and d is the dropout rate applied within the network and before the last layer in each architecture. The grid search evaluates the points $\{(10^{-6}, 0), (10^{-5}, 0), (10^{-6}, 0.2), (10^{-4}, 0.2)\}$, i.e., spanning from a low regularization pressure to a strong one.

Figure 3 visualizes the distribution of obtained validation scores of the tested hyperparameters. Most notably, the convolutional architectures tend to have lower variance, i.e., tolerate a wider set of hyperparameters. Moreover, the individual best scores of the models are all in a relatively small range, i.e., between 0.2 and 0.3, demonstrating the necessity of a proper hyperparameter tuning process.

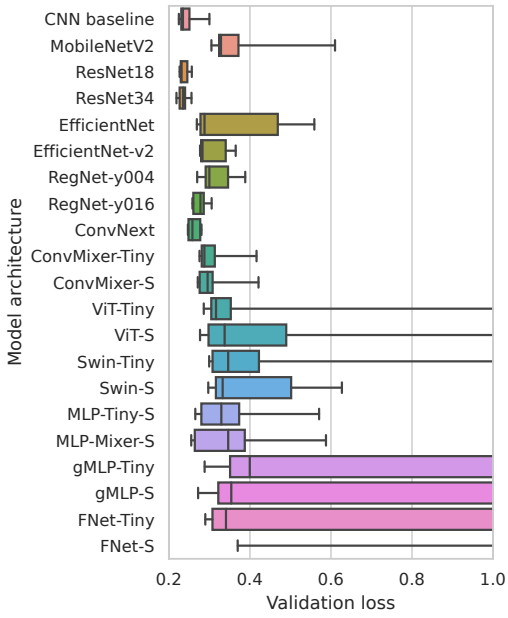

Fig. 3. Box-plot showing the validation loss distribution for the different hyperparameters tested for each model. The whiskers represent the minimum/maximum, the box the 0.25, 0.5, and 0.75 quantiles of the values.

IV. Experimental Results

In this section, we describe our findings on a series of end-to-end imitation learning from offline training to online testing tasks, ranging from autonomous driving to many arcade learning environment games.

A. End-to-end autonomous driving

Our first experiment concerns learning the end-to-end control of an autonomous vehicle. We collect data on a full-scale autonomous vehicle with a 30Hz BFS-PGE-23S3C-CS RGB Camera with resolution 960 $\times$ 600 and 130°. Each image is temporally synchronized with the steering angle estimated by a differential GPS and an IMU to construct a training pair. The dataset consists of roughly 5-hour driving data collected in different times of the day, different road types, and different seasons, e.g., see Figure 2. Among all variations, we use summer and winter data for training set with a fraction put aside for (in-distribution) testing and leave fall, spring, and night data for (out-of-distribution) evaluation. For image preprocessing, we perform center cropping as we focus on lane tracking in this work, and we adopt data augmentation, including randomization in brightness, saturation, hue, and gamma, finally followed by per-image normalization. To improve over compounding error generated by imitation learning, we use Guided Policy Learning (GPL) [79] to generate off-orientation training data and teach the policy how to recover from such scenarios [44]. To test our model in a closed-loop setting, we leverage a high-fidelity data-driven simulator [44] that can be built upon the collected dataset. Trained agents are placed within these simulated environments and are capable of perceiving novel viewpoints in the scene as they execute their policies. The resolution of the input images is 48-by-160 pixels, and all models are trained for 600k steps with a batch size of 64. The settings in terms of input resolution and learning rate schedule were empirically calibrated on the baseline model to achieve acceptable driving performance.

For each model and data condition pair (summer, winter, fall, spring, and night), we run a total of 200 evaluations. An evaluation consists of the model controlling the vehicle's steering with a constant velocity until the vehicle either crashes (i.e., leaves the road) or a certain distance has been driven. We report the number of evaluations that terminated with a crash as our performance metric, with an optimal model counting zero crashes.

The result in Table I shows the number of crashes for the five different environmental conditions and the aggregated counts over all 1000 evaluation runs. The first two columns show that no crash was observed for any model in the summer and winter conditions. Note that data used for the summer and winter simulation does not overlap with the training data, they only share the season of their data collection process. In the out-of-distribution environment

TABLE I

END-TO-END AUTONOMOUS DRIVING. NUMBERS SHOW THE NUMBER OF EXPERIMENT RUNS THAT CRASHED BEFORE SUCCESSFUL TERMINATION. THE VALUE IN PARENTHESES SHOWS THE PERCENTAGE. THE EXPERIMENTS FOR EACH MODEL IN EACH COLUMN ARE REPEATED 200 TIMES. TOTAL NUMBER OF EXPERIMENTS=21000 (1000 INFERENCE EXPERIMENTS FOR EACH MODEL IN 5 DIFFERENT ENVIRONMENTS).

Model	**Number of crashes**					
Condition:	Summer	Winter	Fall	Spring	Night	All
Seen in training:	(in-distribution)		(out-of-distribution)			
CNN baseline	0 (0%)	0 (0%)	**13 (7%)**	**24 (12%)**	108 (54%)	**145 (15%)**
MobileNetV2	0 (0%)	0 (0%)	28 (15%)	48 (24%)	83 (42%)	159 (16%)
ResNet18	0 (0%)	0 (0%)	64 (32%)	57 (29%)	118 (59%)	239 (24%)
ResNet34	0 (0%)	0 (0%)	59 (30%)	46 (23%)	115 (58%)	220 (22%)
EfficientNet	0 (0%)	0 (0%)	33 (17%)	45 (23%)	105 (53%)	183 (19%)
EfficientNet-v2	0 (0%)	0 (0%)	23 (12%)	**39 (20%)**	99 (50%)	161 (17%)
RegNet-y004	0 (0%)	0 (0%)	**18 (9%)**	44 (22%)	80 (40%)	**142 (15%)**
RegNet-y016	0 (0%)	0 (0%)	**12 (6%)**	48 (24%)	96 (48%)	156 (16%)
ConvNext	0 (0%)	0 (0%)	**16 (8%)**	49 (25%)	75 (38%)	**140 (15%)**
ConvMixer-Tiny	0 (0%)	0 (0%)	30 (15%)	58 (29%)	110 (56%)	198 (20%)
ConvMixer-S	0 (0%)	0 (0%)	25 (13%)	63 (32%)	111 (56%)	199 (20%)
ViT-S	0 (0%)	0 (0%)	21 (11%)	**40 (20%)**	70 (35%)	**131 (14%)**
ViT-Tiny	0 (0%)	0 (0%)	22 (11%)	67 (34%)	118 (59%)	207 (21%)
Swin-S	0 (0%)	0 (0%)	**13 (7%)**	55 (28%)	65 (33%)	**133 (14%)**
Swin-Tiny	0 (0%)	0 (0%)	23 (12%)	65 (33%)	99 (50%)	187 (19%)
MLP-Mixer-S	0 (0%)	0 (0%)	24 (12%)	58 (29%)	**48 (24%)**	**130 (13%)**
MLP-Tiny-S	0 (0%)	0 (0%)	**1 (1%)**	63 (32%)	110 (56%)	174 (18%)
gMLP-Tiny	0 (0%)	0 (0%)	**9 (5%)**	48 (24%)	123 (62%)	180 (18%)
gMLP-S	0 (0%)	0 (0%)	**0 (0%)**	57 (29%)	**31 (16%)**	**88 (9%)**
FNet-S	0 (0%)	0 (0%)	48 (24%)	71 (36%)	**54 (27%)**	173 (18%)
FNet-Tiny	0 (0%)	0 (0%)	29 (15%)	63 (32%)	133 (67%)	225 (23%)
Bold threshold			$\leq$10%	$\leq$ 20%	$\leq$30%	$\leq$15%

conditions, no model was able to maneuver the vehicle across all 600 runs successfully. The best performing model, the gMLP-S had no crash when simulated in fall, but a significant crash rate of 29% and 16% in the spring and night conditions, respectively. Figure 1 contrasts the offline performance measured by the validation loss on the x-axis with the online performance measured by crash likelihood on the y-axis. When comparing the convolutional neural network with the patch-based architectures, no significant discrepancy is observed.

The results in Table I show a weak trend of over-parametrization helping generalization. In particular, the larger variants of the ResNet, ViT, Swin-Transformer, MLP-Mixer, gMLP, and FNet architectures outperformed their smaller counterparts. Consequently, our experiments indicate that the neural scaling laws and the advantage of over-parametrized models [15] apply to the Open-loop to closed-loop generalization as well.

B. Arcade learning environment

As the performance of neural networks is tasks-specific, diverse evaluations are necessary to draw significant conclusions. Here, we study the offline-online generalization gap of vision architectures on a total of 4 different IL tasks of the Arcade Learning Environment (ALE) [16].

The training data for the imitation learning setup are generated by an expert policy in the form of a deep Q-network (DQN) [83] from the stable-baselines3 repository [84]. Specifically, we collect the observations and the corresponding suggested action of DQN as labels for the behavior cloning. As regularization, we inject noise into the closed-loop interaction of the data collection process by overwriting 10% of the DQN's actions with randomly sampled ones. Note that the random actions are only used for driving exploration; the training data contain no random actions.

We collect a total of 2 million data samples for each task, which we split into a training and a validation set with a ratio of 85%:15% of non-overlapping trajectories. We preprocess the observations by converting the images by rescaling them to gray-scale with a resolution of 84-by-84 and stacking the four most recent frames, i.e., a technique known as *deepmind* processor in the literature. For each architecture, we use the tuned hyperparameters from before and train for 200k steps.

Our performance measure is the episodic return, i.e., the non-discounted sum of rewards, when deploying the networks in the closed-loop for a total of 20 episodes. We repeat this experiment, including the training process, for three different random seeds. The results are shown in Table II. For each environment, we highlight scores that are within 1/5 of the standard deviation of the best-performing model in bold. We also rank the different network architectures by counting for how many environments they are bolding threshold. The results show that no single model consistently outperforms other architectures and that overall, each model could achieve an excellent closed-loop performance (with only a few outliers). Nonetheless, we observe a minor trend that conv networks achieve a top (bolded) performance more frequently compared to patch-based networks.

TABLE II

IN-DISTRIBUTION ONLINE TESTING OF THE ARCADE LEARNING ENVIRONMENTS. NUMBERS REPORT THE AVERAGE EPISODE RETURN FOR 21 MODELS ON 13 DIFFERENT GAMES; EACH MODEL IS TRAINED THREE TIMES AND TESTED 20 TIMES AT INFERENCE. TOTAL OF 16380 EXPERIMENTS.

Model	*Alien*	*Beamrider*	*Qbert*	*Seaquest*	*Ranking*
CNN baseline	1388	6541	12451	**4574**	3
MobileNetV2	1468	6282	4124	784	2
ResNet18	1454	**6818**	**12986**	**4483**	**6**
ResNet34	**1547**	6514	12614	**4369**	**6**
EfficientNet	1408	6406	**12884**	4106	**5**
EfficientNet-v2	1475	6476	**12785**	4275	**5**
RegNet-y004	1499	**6630**	**12972**	4301	4
RegNet-y016	1444	6459	**12876**	4185	4
ConvNext	1459	6343		4352	2
ConvMixer-Tiny	**1515**	6535	12610	4316	3
ConvMixer-S	1425	6363	12520	4206	0
ViT-S	**1530**	6137	**12649**	**4516**	4
ViT-Tiny	1442	**6684**	12565	**4382**	3
Swin-S	**1538**	6143	**13220**	4122	2
Swin-Tiny	1473	**6630**	12089	4021	2
MLP-Mixer-S	1465	6250	12111	4301	0
MLP-Tiny-S	**1542**	5874	**12738**	3992	2
gMLP-Tiny	**1569**	**6863**	**13167**	4242	4
gMLP-S	1451	6520	12584	4159	2
FNet-S	1443	6361	**13132**	3873	3
FNet-Tiny	1433	6290	**12951**	**4474**	4
Expert policy	1307	6979	12995	3177	
$\pm$ 1/5 std. dev.	$\pm$ 67	$\pm$ 321	$\pm$ 583	$\pm$ 82	

Arcade learning environment under distribution-shift

We also perform an evaluation of how well the models trained on the ALE tasks tolerate a change in the data distribution. In particular, we study the effects of 10 different data modification policies on the closed-loop performance measured in the episodic return of the trained ALE models. The data modification policies are Gaussian blurring with kernel size 3x3 and 5x5, cutting out a 6x6 and 12x12 window in the center of the frame, increasing the brightness by 16 and 32, decreasing the brightness by 16 and 32, and adding uniform noise with a spread of $\pm$ 16 and $\pm$ 32 (pixel values range from 0 to 255). We select the ALE environments *Alien*, *Beamrider*, *QBert*, and *Seaquest* for our evaluation as most models achieve a decent performance on them, thus avoiding skewing the distribution-shift robustness by a difference in in-distribution performance.

The results of the ALE out-of-distribution experiments are shown in Table III. Contrarily to the in-distribution evaluations, we observe a minor trend of the patch-based models performing slightly better than convolutional neural network architectures in environments with shifted distributions.

V. CONCLUSION

In this work, we studied the open-loop to closed-loop causality gap where a neural network is trained offline on labeled data but deployed in a closed-loop system where the decisions of the network affect the next observation, a setting often found in control and robotics. Our study focused on image data and modern vision models. Specifically, we compared convolutional neural networks with Vision Vransformers with recently proposed patch-based architectures. Our results showed that under a proper training setting any architecture can handle the open-loop to closed-loop causality gap. We also showed that changing the environment, i.e., a distribution shift, has catastrophic consequences on

Our results indicate that overparametrization can help open-loop to closed-loop generalization to some extent but more fundamental approaches than advances in vision processing architectures are necessary to handle the causality gap and a possible change in the data distribution simultaneously. A potential future research direction is to incorporate discrete [85], [68] and continuous recurrent mechanisms [86], [87], [88], and structural state-space models [89], [90], [91] in the design of vision networks.

TABLE III

MEAN EPISODE RETURN OF THE OOD DEPLOYMENT OF THE ARCADE MODELS. VALUES WITHIN $1/5$ OF THE STANDARD DEVIATION OF THE BEST PERFORMING MODEL ARE HIGHLIGHTED IN BOLD.

Model	Alien	Beamrider	Qbert	Seaq.
CNN baseline	877	4889	8137	2906
MobileNetV2	922	4723	6360	2218
ResNet18	1024	**5486**	7006	3219
ResNet34	990	5167	7140	3465
EfficientNet	770	4648	6665	2922
EfficientNet-v2	921	4917	7182	3399
RegNet-y004	891	**5329**	7001	3251
RegNet-y016	787	4997	5847	2886
ConvNext	**1292**	**5443**	6970	3505
ConvMixer-Tiny	1098	5101	7778	1938
ConvMixer-S	1153	**5549**	**8412**	2428
ViT-S	1025	4353	5983	3177
ViT-Tiny	1044	4618	5654	2280
Swin-S	1211	**5268**	8023	2812
Swin-Tiny	1188	5203	6512	2670
MLP-Mixer-S	1082	4925	6384	3030
MLP-Tiny-S	1141	4249	5822	3060
gMLP-Tiny	1143	**5580**	7190	2742
gMLP-S	1192	5114	7017	**3586**
FNet-S	1080	**5330**	**9180**	3166
FNet-Tiny	1043	**5318**	**8479**	3262

ACKNOWLEDGMENT

This work was partially supported in parts by the ERC-2020-AdG 101020093. Additionally, it was partially sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. This work was further supported by The Boeing Company and the Office of Naval Research (ONR) Grant N00014-18-1-2830.

REFERENCES

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2020.

[2] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[3] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[4] M. Lechner, R. Hasani, Z. Babaiee, R. Grosu, D. Rus, T. A. Henzinger, and S. Hochreiter, "Entangled residual mappings," *arXiv preprint arXiv:2206.01261*, 2022.

[5] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *arXiv preprint arXiv:2105.01601*, 2021.

[6] A. Trockman and J. Z. Kolter, "Patches are all you need?" *arXiv preprint arXiv:2201.09792*, 2022.

[7] H. Liu, Z. Dai, D. R. So, and Q. V. Le, "Pay attention to mlps," *arXiv preprint arXiv:2105.08050*, 2021.

[8] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon, "Fnet: Mixing tokens with fourier transforms," *arXiv preprint arXiv:2105.03824*, 2021.

[9] A. Amini, I. Gilitschenski, J. Phillips, J. Moseyko, R. Banerjee, S. Karaman, and D. Rus, "Learning robust control policies for end-to-end autonomous driving from data-driven simulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1143–1150, 2020.

[10] W. Xiao, R. Hasani, X. Li, and D. Rus, "Barriernet: A safety-guaranteed layer for neural networks," *arXiv preprint arXiv:2111.11277*, 2021.

[11] W. Xiao, T.-H. Wang, M. Chahine, A. Amini, R. Hasani, and D. Rus, "Differentiable control barrier functions for vision-based end-to-end autonomous driving," *arXiv preprint arXiv:2203.02401*, 2022.

[12] S. Paul and P.-Y. Chen, "Vision transformers are robust learners," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 2071–2081.

[13] M. M. Naseer, K. Ranasinghe, S. H. Khan, M. Hayat, F. Shahbaz Khan, and M.-H. Yang, "Intriguing properties of vision transformers," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[14] Y. Bai, J. Mei, A. L. Yuille, and C. Xie, "Are transformers more robust than cnns?" *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[15] S. Bubeck and M. Sellke, "A universal law of robustness via isoperimetry," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 811–28 822, 2021.

[16] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *Journal of Artificial Intelligence Research*, vol. 47, pp. 253–279, 2013.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[18] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[19] M. Lechner, R. Hasani, R. Grosu, D. Rus, and T. A. Henzinger, "Adversarial training is not ready for robot learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4140–4147.

[20] Y. Fu, S. Zhang, S. Wu, C. Wan, and Y. Lin, "Patch-fool: Are vision transformers always robust against adversarial perturbations?" in *International Conference on Learning Representations*, 2021.

[21] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[24] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[25] ——, "Efficientnetv2: Smaller models and faster training," in *International conference on machine learning*. PMLR, 2021, pp. 10 096–10 106.

[26] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 428–10 436.

[27] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 976–11 986.

[28] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[29] J. Ho and S. Ermon, "Generative adversarial imitation learning," *arXiv preprint arXiv:1606.03476*, 2016.

[30] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *in Proc. 17th International Conf. on Machine Learning*. Citeseer, 2000.

[31] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 661–668.

[32] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[33] Y. Schroecker and C. Isbell, "State aware imitation learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 2915–2924.

[34] H. Le, N. Jiang, A. Agarwal, M. Dudík, Y. Yue, and H. Daumé, "Hierarchical imitation and reinforcement learning," in *International conference on machine learning (ICML)*. PMLR, 2018, pp. 2917–2926.

[35] S. Desai, I. Durugkar, H. Karnan, G. Warnell, J. Hanna, P. Stone, and A. Sony, "An imitation from observation approach to transfer learning with dynamics mismatch," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[36] Y. Duan, M. Andrychowicz, B. C. Stadie, J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba, "One-shot imitation learning," *arXiv preprint arXiv:1703.07326*, 2017.

[37] T. Yu, P. Abbeel, S. Levine, and C. Finn, "One-shot hierarchical imitation learning of compound visuomotor tasks," *arXiv preprint arXiv:1810.11043*, 2018.

[38] N. Baram, O. Anschel, I. Caspi, and S. Mannor, "End-to-end differentiable adversarial imitation learning," in *International conference on machine learning (ICML)*. PMLR, 2017, pp. 390–399.

[39] Y.-H. Wu, N. Charoenphakdee, H. Bao, V. Tangkaratt, and M. Sugiyama, "Imitation learning from imperfect demonstration," in *International conference on machine learning (ICML)*. PMLR, 2019, pp. 6818–6827.

[40] M. Sun and X. Ma, "Adversarial imitation learning from incomplete demonstrations," *arXiv preprint arXiv:1905.12310*, 2019.

[41] P. A. Ortega, M. Kunesch, G. Delétang, T. Genewein, J. Grau-Moya, J. Veness, J. Buchli, J. Degrave, B. Piot, J. Perolat, *et al.*, "Shaking the foundations: delusions in sequence models for interaction and control," *arXiv preprint arXiv:2110.10819*, 2021.

[42] M. Janner, Q. Li, and S. Levine, "Reinforcement learning as one big sequence modeling problem," *arXiv preprint arXiv:2106.02039*, 2021.

[43] D. Ha and J. Schmidhuber, "World models," *arXiv preprint arXiv:1803.10122*, 2018.

[44] A. Amini, T.-H. Wang, I. Gilitschenski, W. Schwarting, Z. Liu, S. Han, S. Karaman, and D. Rus, "Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2419–2426.

[45] A. Brunnbauer, L. Berducci, A. Brandstätter, M. Lechner, R. Hasani, D. Rus, and R. Grosu, "Latent imagination facilitates zero-shot transfer

in autonomous racing," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7513–7520.

[46] T. D. Ullman, E. Spelke, P. Battaglia, and J. B. Tenenbaum, "Mind games: Game engines as an architecture for intuitive physics," *Trends in cognitive sciences*, vol. 21, no. 9, pp. 649–665, 2017.

[47] A. Achille and S. Soatto, "Information dropout: Learning optimal representations through noisy computation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2897–2905, 2018.

[48] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[49] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 262–15 271.

[50] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, *et al.*, "Analysis of representations for domain adaptation," *Advances in neural information processing systems*, vol. 19, p. 137, 2007.

[51] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *International conference on machine learning (ICML)*. PMLR, 2013, pp. 10–18.

[52] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[53] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf, "Domain adaptation with conditional transferable components," in *International conference on machine learning (ICML)*. PMLR, 2016, pp. 2839–2848.

[54] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7167–7176.

[55] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1501–1510.

[56] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," in *International Conference on Learning Representations (ICLR)*, 2018.

[57] H. Wang, Z. He, Z. C. Lipton, and E. P. Xing, "Learning robust representations by projecting superficial statistics out," in *International Conference on Learning Representations (ICLR)*, 2018.

[58] B. Kim, H. Kim, K. Kim, S. Kim, and J. Kim, "Learning not to learn: Training deep neural networks with biased data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9012–9020.

[59] C. Clark, M. Yatskar, and L. Zettlemoyer, "Don't take the easy way out: Ensemble based methods for avoiding known dataset biases," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 4069–4082.

[60] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," *arXiv preprint arXiv:1907.02893*, 2019.

[61] E. Rosenfeld, P. K. Ravikumar, and A. Risteski, "The risks of invariant risk minimization," in *International Conference on Learning Representations (ICLR)*, 2020.

[62] T. Wang, C. Zhou, Q. Sun, and H. Zhang, "Causal attention for unbiased visual recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 3091–3100.

[63] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville, "Out-of-distribution generalization via risk extrapolation (rex)," in *International conference on machine learning (ICML)*. PMLR, 2021, pp. 5815–5826.

[64] Y. Niu, K. Tang, H. Zhang, Z. Lu, X.-S. Hua, and J.-R. Wen, "Counterfactual vqa: A cause-effect look at language bias," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 12 700–12 710.

[65] Z. Yue, T. Wang, Q. Sun, X.-S. Hua, and H. Zhang, "Counterfactual zero-shot and open-set visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 404–15 414.

[66] J. Kim and J. Canny, "Explainable deep driving by visualizing causal attention," in *Explainable and Interpretable Models in Computer Vision and Machine Learning*. Springer, 2018, pp. 173–193.

[67] K. Madan, N. R. Ke, A. Goyal, B. Schölkopf, and Y. Bengio, "Fast and slow learning of recurrent independent mechanisms," in *International Conference on Learning Representations (ICLR)*, 2020.

[68] A. Goyal, A. Lamb, J. Hoffmann, S. Sodhani, S. Levine, Y. Bengio, and B. Schölkopf, "Recurrent independent mechanisms," in *International Conference on Learning Representations*, 2020.

[69] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," *arXiv preprint arXiv:2106.01345*, 2021.

[70] X. Yang, H. Zhang, G. Qi, and J. Cai, "Causal attention for vision-language tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9847–9857.

[71] A. Power, Y. Burda, H. Edwards, I. Babuschkin, and V. Misra, "Grokking: Generalization beyond overfitting on small algorithmic datasets," *arXiv preprint arXiv:2201.02177*, 2022.

[72] R. Hasani, M. Lechner, A. Amini, L. Liebenwein, M. Tschaikowski, G. Teschl, and D. Rus, "Closed-form continuous-depth models," *arXiv preprint arXiv:2106.13898*, 2021.

[73] C. Vorbach, R. Hasani, A. Amini, M. Lechner, and D. Rus, "Causal navigation by continuous-time neural networks," *arXiv preprint arXiv:2106.08314*, 2021.

[74] M. Lechner, R. Hasani, A. Amini, T. A. Henzinger, D. Rus, and R. Grosu, "Neural circuit policies enabling auditable autonomy," *Nature Machine Intelligence*, vol. 2, no. 10, pp. 642–652, 2020.

[75] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[76] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[77] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[78] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[79] S. Levine and V. Koltun, "Guided policy search," in *International conference on machine learning (ICML)*. PMLR, 2013, pp. 1–9.

[80] R. Hasani, M. Lechner, A. Amini, D. Rus, and R. Grosu, "Liquid time-constant networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 9, pp. 7657–7666, May 2021.

[81] A. Krogh and J. Hertz, "A simple weight decay can improve generalization," *Advances in neural information processing systems*, vol. 4, 1991.

[82] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[83] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[84] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

[85] R. Hasani, A. Amini, M. Lechner, F. Naser, R. Grosu, and D. Rus, "Response characterization for auditing cell dynamics in long short-term memory networks," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[86] M. Lechner, R. Hasani, M. Zimmer, T. A. Henzinger, and R. Grosu, "Designing worm-inspired neural networks for interpretable robotic control," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 87–94.

[87] M. Lechner and R. Hasani, "Learning long-term dependencies in irregularly-sampled time series," *arXiv preprint arXiv:2006.04418*, 2020.

[88] R. Hasani, M. Lechner, A. Amini, D. Rus, and R. Grosu, "The natural lottery ticket winner: Reinforcement learning with ordinary neural circuits," in *Proceedings of the 2020 International Conference on Machine Learning*. JMLR. org, 2020.

[89] M. Lechner, R. Hasani, D. Rus, and R. Grosu, "Gershgorin loss stabilizes the recurrent neural network compartment of an end-to-end robot learning scheme," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5446–5452.
[90] A. Gu, K. Goel, and C. Re, "Efficiently modeling long sequences with structured state spaces," in *International Conference on Learning Representations*, 2021.
[91] R. Hasani, M. Lechner, T.-H. Wang, M. Chahine, A. Amini, and D. Rus, "Liquid structural state-space models," *arXiv preprint arXiv:2209.12951*, 2022.