Are All Vision Models Created Equal? A Study of the Open-Loop to Closed-Loop Causality Gap

Mathias Lechner
MIT
mlechner@mit.edu

Ramin Hasani
MIT
rhasani@mit.edu

Alexander Amini
MIT
amini@mit.edu

Tsun-Hsuan Wang
MIT
tsunw@mit.edu

Thomas A. Henzinger
Institute of Science and Technology Austria (ISTA)
tah@ist.ac.at

Daniela Rus
MIT
rus@mit.edu

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 patch-based networks, have been extensively tested on offline image classification and regression tasks. In this paper, we study these vision architectures with respect to the open-loop to closed-loop causality gap, i.e., offline training followed by an online closed-loop deployment. This causality gap emerges in end-to-end 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 models perform indistinguishably well on in-distribution deployment, resolving the causality gap. In situation 2, We observe that the causality gap disrupts performance regardless of the choice of the model architecture. Our results imply that the causality gap can be solved in situation one with our proposed training guideline with *any* modern network architecture, whereas achieving out-of-distribution generalization (situation two) requires further investigations, for instance, on data diversity rather than the model architecture.

1 Introduction

A tremendous number of advanced deep learning models have been proposed to perform competitively in end-to-end perception-to-control autonomous driving tasks. For example, patch-based vision architectures such as Vision Transformer (ViT) [12] have shown to be competitive with models based on convolutional neural networks (CNNs) [15, 33] in computer vision applications for which CNNs were the predominant choice. A very recent line of research, namely the MLPMixer [60], and ConvMixer [61] 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) [37] or replacing self-attention sublayer with an unparameterized Fourier Transform [34] can outperform ViT.

These proposals are largely tested in offline settings where the output decisions of the network do not change the next 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

Machine Learning for Autonomous Driving Workshop at the 36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, USA.

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

network actions affect next input observations, such as in imitation learning tasks. Imitation learning agents typically suffer from a causality gap arising from the transfer of models from open-loop training to closed-loop testing. In this paper, we focus on investigating this gap for end-to-end autonomous steering of a vehicle in a systematic way.

In this paper, we design an end-to-end autonomous driving (AD) imitation learning experiment to assess the performance of various advanced vision models in handling the open-loop training to closed-loop testing causality gap. In particular, we leverage the photorealistic AD simulation platform called VISTA [2] for closed-loop testing. Moreover, we evaluate the models in 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 a variety of models requires us to ensure fairness and proper evaluation of the effectiveness of different model architectures. To this end, we validate that all baseline models are trained to their best capability given the same decent amount of hyperparameter optimization budget under a controlled training pipeline.

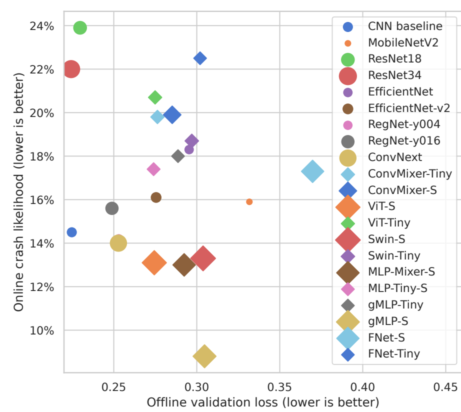

Figure 1: Online deployment vs. offline training causality gap in perspective. Marker size is linearly proportional to the number of trainable parameters.

Counterintuitively and in contrast to the recently reported results [47, 43, 6], 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 the causality gap certainly affects the performance of models, again, almost regardless of the choice of their architecture. These findings suggest the rethinking of the emphasis on the choice of popular models such as Transformers over CNNs, as other factors such as proper training setup, augmentation strategies, and data diversity play a more important role in generalization in and out of distribution.

2 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.

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 (LR) 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 [48, 20, 64].
3. We deploy a custom staircase LR decay schedule that decreases the LR 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 [29], Batch-Normalization [25], and Layer-Normalization [5] modules initialized properly.
5. We replace the standard Adam optimizer with AdamW [40], 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 [35].

We compare a total of 21 different advanced models, including nine modern convolutional networks and 12 modern patch-based architectures. A full description of the architectures and baseline CNN network can be found in Appendix A.

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 [30, 57, 40]. 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.

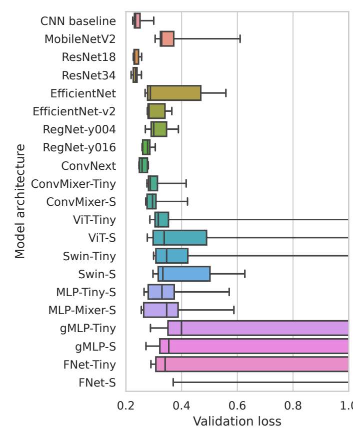

Figure 3: Box-plot showing the validation loss distribution for the different hyperparameters tested for each model. The whiskers represent the minimum/maximum, the box the 0.25, 0.5, and 0.75 quantiles of the values.

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.

3 Experimental Results

Our 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 960×600 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) [35] to generate off-orientation training data and teach the policy how to recover from such scenarios [3]. To test our model in a closed-loop

Table 1: End-to-end autonomous driving. Numbers show the number of experiment runs that crashed before successful termination. The number in parentheses shows the percentage. The experiments for each model in each column are repeated 200 times. The 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$5%	$\leq$ 20%	$\leq$30%	

setting, we leverage a high-fidelity data-driven simulator [3] 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.

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 1 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 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 network with the patch-based architectures, no significant discrepancy is observed.

4 Conclusion

We studied the open-loop to closed-loop causality gap in autonomous driving, where a neural network is trained offline on labeled image data but deployed in a closed-loop system. Specifically, we compared convolutional neural networks with recently proposed patch-based architectures. Our results showed that if properly trained, any architecture can handle the open-loop to closed-loop causality gap, connecting to the observation made in the literature that patch-based architectures are not necessarily more robust than convolutional architectures [14]. We also showed that a change in the data distribution can have catastrophic consequences on the closed-loop generalization.

References

[1] Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2897–2905, 2018.

[2] Alexander Amini, Igor Gilitschenski, Jacob Phillips, Julia Moseyko, Rohan Banerjee, Sertac Karaman, and Daniela Rus. Learning robust control policies for end-to-end autonomous driving from data-driven simulation. *IEEE Robotics and Automation Letters*, 5(2):1143–1150, 2020.

[3] Alexander Amini, Tsun-Hsuan Wang, Igor Gilitschenski, Wilko Schwarting, Zhijian Liu, Song Han, Sertac Karaman, and Daniela 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)*, pages 2419–2426. IEEE, 2022.

[4] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[6] Yutong Bai, Jieru Mei, Alan L Yuille, and Cihang Xie. Are transformers more robust than cnns? *Advances in Neural Information Processing Systems*, 34, 2021.

[7] Nir Baram, Oron Anschel, Itai Caspi, and Shie Mannor. End-to-end differentiable adversarial imitation learning. In *International conference on machine learning (ICML)*, pages 390–399. PMLR, 2017.

[8] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

[9] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*, 2021.

[10] Christopher Clark, Mark Yatskar, and Luke 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)*, pages 4069–4082, 2019.

[11] Siddharth Desai, Ishan Durugkar, Haresh Karnan, Garrett Warnell, Josiah Hanna, Peter Stone, and AI Sony. An imitation from observation approach to transfer learning with dynamics mismatch. *Advances in Neural Information Processing Systems*, 33, 2020.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2020.

[13] Yan Duan, Marcin Andrychowicz, Bradly C Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. *arXiv preprint arXiv:1703.07326*, 2017.

[14] Yonggan Fu, Shunyao Zhang, Shang Wu, Cheng Wan, and Yingyan Lin. Patch-fool: Are vision transformers always robust against adversarial perturbations? In *International Conference on Learning Representations*, 2021.

[15] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[16] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[17] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations (ICLR)*, 2018.

[18] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International conference on machine learning (ICML)*, pages 2839–2848. PMLR, 2016.

[19] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. In *International Conference on Learning Representations (ICLR)*, 2020.

[20] Ramin Hasani, Mathias Lechner, Alexander Amini, Lucas Liebenwein, Max Tschaikowski, Gerald Teschl, and Daniela Rus. Closed-form continuous-depth models. *arXiv preprint arXiv:2106.13898*, 2021.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15262–15271, 2021.

[23] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *arXiv preprint arXiv:1606.03476*, 2016.

[24] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1501–1510, 2017.

[25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[26] Michael Janner, Qiyang Li, and Sergey Levine. Reinforcement learning as one big sequence modeling problem. *arXiv preprint arXiv:2106.02039*, 2021.

[27] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo 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)*, pages 9012–9020, 2019.

[28] Jinkyu Kim and John Canny. Explainable deep driving by visualizing causal attention. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 173–193. Springer, 2018.

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[30] Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

[31] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International conference on machine learning (ICML)*, pages 5815–5826. PMLR, 2021.

[32] Hoang Le, Nan Jiang, Alekh Agarwal, Miroslav Dudík, Yisong Yue, and Hal Daumé. Hierarchical imitation and reinforcement learning. In *International conference on machine learning (ICML)*, pages 2917–2926. PMLR, 2018.

[33] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[34] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021.

[35] Sergey Levine and Vladlen Koltun. Guided policy search. In *International conference on machine learning (ICML)*, pages 1–9. PMLR, 2013.

[36] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[37] Hanxiao Liu, Zihang Dai, David R So, and Quoc V Le. Pay attention to mlps. *arXiv preprint arXiv:2105.08050*, 2021.

[38] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[39] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[41] Kanika Madan, Nan Rosemary Ke, Anirudh Goyal, Bernhard Schölkopf, and Yoshua Bengio. Fast and slow learning of recurrent independent mechanisms. In *International Conference on Learning Representations (ICLR)*, 2020.

[42] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International conference on machine learning (ICML)*, pages 10–18. PMLR, 2013.

[43] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

[44] Andrew Y Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *in Proc. 17th International Conf. on Machine Learning*. Citeseer, 2000.

[45] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12700–12710, 2021.

[46] Pedro A Ortega, Markus Kunesch, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Joel Veness, Jonas Buchli, Jonas Degrave, Bilal Piot, Julien Perolat, et al. Shaking the foundations: delusions in sequence models for interaction and control. *arXiv preprint arXiv:2110.10819*, 2021.

[47] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2071–2081, 2022.

[48] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

[49] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020.

[50] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34, 2021.

[51] Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations (ICLR)*, 2020.

[52] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

[53] Stéphane Ross, Geoffrey Gordon, and Drew 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*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[54] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[55] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999.

[56] Yannick Schroecker and Charles Isbell. State aware imitation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2915–2924, 2017.

[57] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[58] Mingfei Sun and Xiaojuan Ma. Adversarial imitation learning from incomplete demonstrations. *arXiv preprint arXiv:1905.12310*, 2019.

[59] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[60] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

[61] Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022.

[62] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7167–7176, 2017.

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[64] Charles Vorbach, Ramin Hasani, Alexander Amini, Mathias Lechner, and Daniela Rus. Causal navigation by continuous-time neural networks. *arXiv preprint arXiv:2106.08314*, 2021.

[65] Haohan Wang, Zexue He, Zachary C Lipton, and Eric P Xing. Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations (ICLR)*, 2018.

[66] Tan Wang, Chang Zhou, Qianru Sun, and Hanwang Zhang. Causal attention for unbiased visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3091–3100, 2021.

[67] Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. Imitation learning from imperfect demonstration. In *International conference on machine learning (ICML)*, pages 6818–6827. PMLR, 2019.

[68] Xu Yang, Hanwang Zhang, Guojun Qi, and Jianfei Cai. Causal attention for vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9847–9857, 2021.

[69] Tianhe Yu, Pieter Abbeel, Sergey Levine, and Chelsea Finn. One-shot hierarchical imitation learning of compound visuomotor tasks. *arXiv preprint arXiv:1810.11043*, 2018.

[70] Zhongqi Yue, Tan Wang, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Counterfactual zero-shot and open-set visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15404–15414, 2021.

A 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 [63] on natural language processing (NLP) datasets, [12] introduced the *Vision Transformer* (ViT) by adapting the architecture for computer vision tasks. As transformers operate on a 1-dimensional sequence of vectors, [12] 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. Researchers have analyzed the difference between how CNNs and ViTs process images [50]. Moreover, it has been claimed that vision transforms are much more robust to image perturbations and occlusions [43], as well as be able to handle distribution-shifts [6] better than CNNs. However, more recent works have refuted the robustness claims of vision transformers [14] by showing that ViTs can be less robust than convolutional networks when considering carefully crafted adversarial attacks.

Swin Transformer [38] 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. This hierarchical processing allows it to use smaller patches without exploding the compute and memory footprint of the model.

MLP-Mixer [60] 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, i.e., mixing features and mixing spatial information.

gMLP [37] is another MLP-only vision architecture that differs from the MLP-Mixer by introducing multiplicative spatial gating units between the alternating spatial and feature MLPs. Empirical results [37] show that the gMLP has a better accuracy-parameter ratio than the MLP-Mixer.

FNet [34] 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. Although the authors [34] did not evaluate the model for vision tasks, FNet's similarity to patch-based MLP architectures makes it a natural candidate for vision tasks.

ConvMixer [61] 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 [61]. The authors claim a large part of the performance of MLP and vision transformers can be attributed to the patch-based processing instead of the type of mixing representation [61].

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

ResNet [21] add skip connections that bypass the convolutional layers. This simple modification allows training much deeper networks than a pure sequential composition of layers. Consequently, skips connections can be found in any modern neural network architecture, including patch-based and advanced convolutional models.

MobileNetV2 [54] 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 [59] is an efficient convolutional neural network architecture derived from an automated neural architecture search. The objective of the search is to find a network topology that achieves high performance while simultaneously running efficiently on CPU devices.

EfficientNet-v2 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 [49] 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 [39] 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.

Baseline CNN We compare the advanced network architectures described above with a 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. A global average pooling layer is applied to feature maps of the final convolution to a single vector.

Imitation learning (IL). IL describes learning an agent by expert demonstrations consist of observation-action pairs [55], directly via behavior cloning [23], or indirectly via inverse reinforcement learning [44]. 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 [52, 53], state-aware imitation [56, 32, 11], pre-trained policies through meta-learning [13, 69], min-max optimization schemes [23, 7, 67, 58], and using insights from causal inference [46, 26].

OOD generalization. It is fundamentally challenging for statistical models to tackle OOD problems [1, 36, 22], such as domain adaptation [8, 42, 16, 18, 62], debiasing [24, 17, 65, 27, 10], and even practically more challenging settings where OOD semantics are unlabeled [4, 51, 66, 31]. A large body of recently proposed solutions to OOD generalization, explored causal inference such as causal interventions [66, 46], designing counterfactual schemes [45, 70], and using attention-based models [28, 41, 19, 9, 26, 68]. 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.