Efficient Behavioral Cloning for Cross-Domain Generalization in End-to-End Driving

Anonymous Author(s)
Affiliation
Address
email

Abstract: Vision-based end-to-end driving suffers from domain gaps between simulation data and real-world data. In the application of autonomous driving, diverse real-world data is costly to collect, even more so for edge cases, such as collisions or unpredictable behavior. Models trained on real-world data can benefit from simulated data through transfer learning, but a domain gap stands in the way of high-level generalization, even among real world datasets. Research on vision transformers has shown in recent years that attention-based architectures have better generalization capabilities than CNN-based counterparts. Ideally, vehicles that have learned to drive in one domain should generalize well to unseen domains regardless of differences in lighting, distortions, and coloring; we characterize this task as single-source or limited-source domain generalization. In this paper, we present a novel video augmentation and contrastive learning routine for the behavioral cloning vehicle steering task which takes advantage of vision transformers' invariance to patch order, as well as similar label spaces of control sequences across domains.

Keywords: Autonomous driving, augmentation, contrastive learning, generalization

1 Introduction

Autonomous driving is an application where meaningful learning is crucial for generalizing learned behaviors across domains. Learning across domains is a common problem, useful to current infrastructure and resources, and necessary for utilizing datasets collected from all kinds of environments in the real world. Difficult edge cases such as dangerous driving, adverse weather behavior, or vehicle accidents are easily collected in simulation. Thus, behaviors learned in the simulated domain should also be both meaningful and generalizable to the real world.

In this work, we focus on the task of purely vision-based behavioral cloning steering angle prediction for autonomous vehicles, with the goal being single or limited-source domain generalization. Single or limited-source domain generalization is crucial to driving models for efficient fine-tuning and is indicative of meaningful behavior learning from source data. Domain generalization is most traditionally achieved by involving a mixed bag of data from several domains in training. Driving environments across the world along with small differences in sensors (such as camera distortion) are too diverse to account for in every dataset, but are also inherently similar in dynamics and structure in many ways.

Given the setting of autonomous driving, spatio-temporal visual learning is natural and does not require additional sensors with regard to physical hardware. The temporal dimension should be highly informative for networks to predict steering angle, as the trajectory of previous data can greatly affect the steering decision given two similar images.

Virtually all driving datasets share high-level visual elements, including lane markings, a horizon and sky, and complex scenes with cars, signs, lights, and trees. An effective machine learning

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

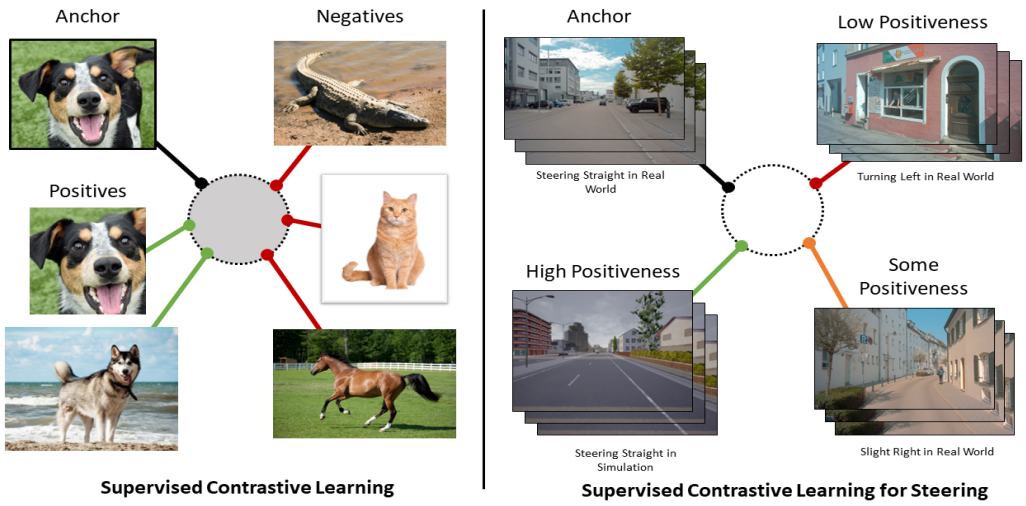

Figure 1: **Contrastive Loss for Steering.** Contrastive loss for behavioral cloning differs from supervised contrastive learning [1] in that positive samples are defined on a continuous similarity metric rather than dicretely "positive" or "negative". "Positiveness" is determined by cosine similarity between steering label vectors. This can be extended to behavioral cloning problem in general, where a temporal sequence of control labels is accessible.

system should benefit from the fact that the co-occurrence of these objects is nearly the same across datasets, even if the pixel representations are different. From this observation, we hypothesize that augmenting input video sequences should improve learning generalization specifically for driving applications, and that common label spaces can be leveraged with contrastive learning to further improve cross-domain finetuning.

We propose an improved learning scheme which enforces learning of out-of-distribution data by grouping samples based on common control sequence labels and diversifying input image sequences through patch-based augmentation. Patch-based augmentation involves randomly changing brightness, saturation, contrast, and hue across frame-level patches in a single image sequence. This augmentation leverages the property that vision transformers tokenize input into patches before passing it into the network, and that learning is invariant to the order of patches. In contrast, convolution-based networks may require perturbation continuity on the frame level.

Our model shows similar or improved performance compared to other methods involving data beyond the source domain. In contrast to alternatives in distillation or domain adaptation, our contrastive training routine for robot behavioral cloning requires data from only one domain at minimum, and also does not require another network for adversarial feature learning. Our method is also simple to implement, with no added complexity to existing models, and only a fraction amount of added wall-time on training compared to adversarial methods. Unseen test domains include other simulation data with visual differences and real-world data, which are not involved during phase one training. We also show the effect of spatial and temporal transfer in an ablation study.

Since spatio-temporal transformers have not been explored in end-to-end driving, to the best of our knowledge, we also provide results on video-input architectures on the end-to-end driving task to provide performance baselines for our method.

In summary, the main contributions of this paper include:

1. A modified supervised contrastive learning scheme for behavioral cloning robot learning applications such as learn-to-steer, which leverages continuous similarity between samples via control sequence labels.

2. A patch-based video augmentation scheme for vision transformer network architectures, which introduces different coloration perturbation combinations in one sample more efficiently in training than sample-level perturbations.

3. Results and ablation study on the performance of transformer-based end-to-end driving models, which had previously not been tested

2 Related Works

Spatio-temporal Vision Transformers.

A natural extension of vision transformers is transformers modeling spatio-temporal dependencies from 3-dimensional image data [2, 3, 4, 5, 6]. The self-attention mechanism in transformers, which has an $O(N^2)$ computational complexity for n tokens, becomes more pronounced with an extra temporal dimension. Thus works exploring transformers in spatio-temporal settings are faced with mitigating computational complexity as well as achieving higher performance [7]. Arnab et al. propose a video vision transformer (ViViT) which, similarly to traditional vision transformers, uses vanilla transformer-based architecture on video data and explores different methods of modeling efficient space-time relationships through attention schemes or factorized architectures [8]. Around the same time, Bertasius et al. proposed TimeSformer, which captures spatio-temporal dependencies within data via different attention schemes [9]. Similarly, XViT also explores different attention schemes across the temporal dimension [10]. We benchmark these architectures for the end-to-end steering task, which is relatively unexplored and fundamentally different from perception tasks in that the label describes a more abstract notion of control.

Video Augmentation. While image augmentation become a practical standard for spatial recognition tasks, sophisticated video-based augmentation is much less explored due to challenges introduced by temporal continuity. Augmentations changing the pixel position of objects in the scene must be consistent across all frames, and label manipulation is not as straightforward as in image classification tasks. Video augmentation can be performed naively via popular image augmentation methods such as MixUp, Cutout, CutMix, RandAug, and AugMix [11, 12, 13, 14, 15] by extending augmentations to every frame in one sample and keeping the mixing of samples consistent to frame index.

CutMix was extended to videos in VideoMix [16] in this fashion, where video samples were mixed similarly to CutMix, except video cuboids were inserted into samples instead of 2-dimensional image patches. Budvytis et al. introduced a large-scale method for video augmentation on segmentation tasks for driving [17], in which hand-labeled samples at certain keyframes were used to infer pixel-level labels for in-between frames for video. In contrast, our work handles a more simple and generalizable problem for data augmentation such that label inference is not necessary. Sun et al. [18] also leverages video augmentation in the context of contrastive learning, but does so by explicitly providing augmentation parameterizations to the learner. Our method involves augmentation on the sample level, where additional augmentation encoders are not involved. Our method also aims to address domain generalization, whereas the goal in this work is same-domain generalization.

Contrastive Learning for Spatio-temporal Control Contrastive learning is a popular method first introduced in [19] which, based on some grouping criteria, "pulls“ similar samples in the feature space closer and "pushes“ different samples farther apart, creating distinctive decision boundaries. Khosla et al. extended this to leverage class labels as grouping criteria, making contrastive learning supervised and explicit to label class [1]. This had further been extended to video [20, 21], where sample similarity is generated via mixing of videos or by quantifying optical flow across images. While these works explore video augmentation without labels, we leverage temporal control labels for our sample similarity criterion, and focus specifically on control tasks. Sermanet et al. [22] explore contrastive learning for control by leveraging features which do not change across viewpoint, but do change across time. These features are then used as part of the reward function in reinforcement learning. In contrast, our paper formulates contrastive learning for behavioral cloning methods.

3 Method

Transfer learning is most commonly achieved via fine tuning of a pre-trained model. Applications such as autonomous driving can benefit from transfer learning for cross-domain learning. Our method can be considered an enhanced fine-tuning method which involves iterative optimization in the target domain as well as a contrastive objective for behavioral cloning settings.

For backbone, we primarily use the architecture design from model 2 of the Video Vision Transformer (ViViT) by Arnab et al. [8], where spatial and temporal encoders are factorized and showed better performance against factorized self-attention and factorized dot-product on video-related tasks. To motivate the need for such transfer learning, we formally define the Simulation-to-Real World (Sim2Real problem for autonomous driving tasks.

Figure 2: **Patch augmentation visualization.** An example of patch augmentation on a 224x224 image sequence with patch size 16 on a sample from Waymo dataset. Top row: augmented sample. Bottom row: original sample.

3.1 Cross-domain Learning Use Case: Sim2Real

Sim2Real describes the problem of resolving gaps in learning between a simulated environment versus the real world. Zhao et al. describe the limitations of collecting real-world data in their survey [23] for deep reinforcement learning in robotics; the same problems apply to imitation learning problems with deep neural networks, where simulated data may be cheap and accessible and real-world data may be costly or unrealistic to obtain. For example, while there is an abundance of real-world driving data, many datasets describe proper driving in suburban, non-congested scenarios and lack substantial representation of scenarios requiring aggressive maneuvers, defensive driving, or avoiding collisions and pedestrians. It is worth noting that while representation in driving data is a challenge, this paper addresses a technique which improves understanding across domains at no additional cost to model complexity or extra data.

Image-based driving data typically represents the front dashboard view of a vehicle. While other methods for autonomous driving may consider different views and modalities, we consider the problem of single image modality for end-to-end driving. For front-view images, the distributions of spatial embeddings learned by the model may be substantially different between simulated and real domains. However, the optical flow, or motion of individual pixels in images, will be similar across both domains, as the vehicle holding the object perform similar movements across the same degrees of freedom.

3.2 Training Procedure

We take advantage of this intuition for our method, where visually observed data such as images may be different across domains, but the labels representing control sequences are within the same space across domains. We present enhancements to finetuning for transformer-based spatio-temporal models through 1) an additional contrastive objective based on spatio-temporal steering angle labels, and 2)

patch-level data augmentation. For our experiments, the source model is identical in architecture to the target model, and no layers are frozen. First, the source model is trained on simulated data in a straightforward manner. Then, the target model is initialized with the source model weights, and resumes training on real-world data. This method is generalizable to fine-tuning independent of the source model, as long as weight initialization is possible.

3.3 Contrastive Learning for Behavioral Cloning

We implement supervised contrastive learning [24] for the end-to-end driving setting to drive samples with similar label sequences closer together in the model's feature space. Similarly to the motivations from [24], we use label information with contrastive learning to capture vehicle motion across frames. We take advantage of the fact that vehicles will move in similar ways regardless of visual domain. For example, a vehicle turning right in simulation will have the similar labels as a vehicle turning right in the real world, given they are going the same speed. Intuitively, the steering labels from both simulation and the real world will have the similar properties and distributions. We reiterate the original supervised contrastive loss below:

$$\mathcal{L} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} log \frac{exp(z_i \cdot z_p)/\tau}{\sum_{a \in A(i)} exp(z_i \cdot z_a)/\tau} \tag{1}$$

where index i is the anchor, index p is a positive sample, z is a feature vector, and $P(i) \equiv \{p \in A(i) : \tilde{y}_p = \tilde{y}_i\}$ is the set of indices of all positives in the batch besides i. In this case, a positive index p is defined when the class label of that sample is equivalent to that of the anchor.

Another thing consider is that "positives" and "negatives" here are binary, and can be defined on the output features using a binary mask; such mask results in the definition of P. Instead of considering a mask of purely 1's and 0's, we consider a weight matrix instead of values between 0 and 1. This weight matrix is determined by the pairwise dot product of output features for each sample in the batch. Thus, instead of having a set of positives P, we define "positiveness" in contrast with vector similarity between temporal labels. Our temporally-supervised contrastive loss function is defined below:

$$w_{i,a} = cos(\tilde{y}_i, \tilde{y}_a) = \frac{y_i \cdot y_a}{||y_i||\ ||y_a||} \tag{2}$$

$$\mathcal{L} = \sum_{i \in I} \frac{-1}{|A(i)|} \sum_{a \in A(i)} w_{i,a} log \frac{exp(z_i \cdot z_a)/\tau}{\sum_{a' \in A(i)} exp(z_i \cdot z_{a'})/\tau} \tag{3}$$

3.4 Patch Augmentation of Video Input

In architectures derivative of Vision Transformers (ViT) [25], images are generally split into fixed-sized patches, are linearly embedded, summed with positional embeddings, then fed to the rest of the network. Within the transformer network, pairwise relationships between patches are characterized by self-attention. The self-attention mechanism, along with the rest of the Transformer network, should be invariant to minor changes in coloration at the patch level. Because of this patching mechanism, inherent discontinuities for learning exist at pixels along patch borders. We leverage this to increase the unique perturbations per sample on the patch level to increase diversity of samples in one training iteration.

Patch augmentation is applied similarly to traditional image augmentations, but instead on the patch-level. Patch size for augmentation is equal to the patch size hyperparameter defined for the transformer model. Given an RGB image which is of dimension $H \times W \times 3$ with square patch size P, let $N = H/P$ and $M = W/P$. Then, an image sample will have NM patches. For video input of t frames, the number of patches then becomes tNM. One range parameter between 0 and 1 inclusive is defined for brightness, contrast, saturation, and hue (r_b, r_c, r_s, r_h respectively).

Then, for a probability $p \in [0, 1]$, a patch's brightness, contrast, saturation, and hue are altered by a factor between $max(0, 1 - r)$ and $1 + r$. This is most commonly known as color jitter. Patch augmentation across frames are independent of each other; one patch at one timestep may have a different perturbation than the same patch at the next timestep. While this may seem to degrade learning initially, perturbing patches at same locations with different parameters may encourage learning of meaningful and higher-level semantic features over time, rather than learning domain-specific pixel values.

We choose to limit patch augmentation to coloration to preserve semantic continuity across frames; rotation and cropping by patch can obviously lead to issues in preserving semantic meaning to images.

Compared to traditional data augmentation where one sample produces one perturbed sample, patch augmentation produces more color perturbations per sample by a factor of tNM, or the number of patches. For each training iteration, the learner is then exposed to a more diverse variety of perturbation parameters, which can improve robustness more efficiently.

Table 1: **Comparison Across Transfer Methods**. We compare several knowledge transfer methods against our method across several styles of spatio-temporal learning backbones. "Vanilla" denotes straightforward weight transfer for fine tuning. Overall, our method shows to have similar improvement to existing knowledge transfer methods without the need for adversarial learning, additional parameters, or access to both source and target data. All experiments are trained in the target domain. The "Datasets" column denotes the datasets involved in training, where "S" denotes "Source / CARLA" and "T" denotes "Target / Waymo".

Method	Datasets	Waymo	Audi	SullyChen	Udacity	CARLA
ViViT	T	88.8	75.9	88.8	50.0	66.2
ViViT+Vanilla	S,T	**92.2**	79.2	94.8	52.4	69.8
ViViT+KD	S,T	**92.2**	**79.9**	97.1	52.1	70.2
ViViT+ADDA	S,T	**92.2**	79.7	**97.4**	**52.6**	70.4
ViViT+Ours	T	91.7	78.9	96.1	51.9	**70.5**
ViViT+Vanilla+Ours	S,T	92.1	79.6	97.1	52.4	69.8
TimeS	T	**93.8**	79.8	93.5	48.6	68.6
TimeS+Vanilla	S,T	92.4	**80.3**	92.2	50.7	67.8
TimeS+KD	S,T	92.9	79.2	93.7	**52.9**	69.4
TimeS+ADDA	S,T	92.2	79.7	**95.6**	52.4	**70.2**
TimeS+Ours	T	**93.8**	79.3	91.8	51.0	67.9
TimeS+Vanilla+Ours	S,T	93.5	80.2	93.2	50.5	68.5

Table 2: **Comparison of network performance on learn-to-steer**. We compare backbone architecture performance on the learn-to-steer spatio-temporal prediction task. For each network, we conduct training on the Waymo dataset and compare performance across several test datasets. In terms of performance, CNN-based architectures seem to generalize the best for learn-to-steer, as well as being extremely efficient compared to transformer models.

Network	Type	Waymo	Audi	SullyChen	Udacity	CARLA
NVIDIA+LSTM	3D CNN	92.3	**79.9**	**96.2**	**51.9**	**70.3**
ViT	2D Transformer	90.4	77.9	89.1	51.2	68.0
ViViT (Model 2)	3D Transformer	88.8	75.9	88.8	50.0	66.2
TimeSformer	3D Transformer	**93.8**	79.8	93.5	48.6	68.6
SWin	3D Transformer	88.2	79.8	94.9	51.7	66.5

4 Experiments

Hardware specs. Every experiment is conducted with one Intel(R) Xeon(R) W-2255 CPU, one NVIDIA RTX A4000 GPU, and 16 GB RAM.

Task. For end-to-end driving, we set our task to "learn to steer", similar to that in [26], except in the spatio-temporal setting, where a model learns the mapping $f : X \rightarrow Y$ for a temporal sequence of T images $X \in \mathbb{R}^{T \times H \times W \times 3}$ and a steering angle label $Y \in \mathbb{R}$. We calculate regression accuracy for this task similarly to that of Shen et al. [27], where accuracy is considered w.r.t a threshold τ as $acc_\tau = count(|v_{predicted} - v_{actual}| < \tau)/n$, where n denotes the number of test cases and $v_{predicted}$ and v_{actual} indicate the predicted and ground-truth value, respectively.

Specifically, we focus on the setting for Sim2Real transfer learning, where a source network is pre-trained on a simulated domain, then trained on a real-world domain.

Datasets. For all experiments, we set the source and target domain to data from CARLA simulator [28] and Waymo [29], respectively. Since our goal is to achieve transfer learning of more generalizable features, we test performance of each network on several datasets in both real world and simulated domains in addition to test sets from CARLA and Waymo, which include Audi's A2D2 [30], Honda's HRI Driving Dataset [31], SullyChen's Driving Dataset [32], and Udacity Driving Simulator [33].

Backbones. Our method is generalizable to different kinds of spatio-temporal architectures. We experiment on two different kinds of spatio-temporal Vision Transformer designs: Video Vision Transformer (ViViT) [8], where spatial and temporal information is encoded separately on the encoder level, and TimeSformer [9], where spatial and temporal information is encoded on the attention level within a single encoder block. To the best of our knowledge, since the learn-to-steer task has not been thoroughly explored with spatio-temporal Vision Transformers, we also include baseline results directly comparing performance on different architectures trained on the target Waymo dataset in Table 2.

Training hyperparameters. Learning rate for both discriminator MLPs and vision-based Transformers are set to 0.000001. We use the AdamW optimizer [34] with a weight decay of 0.01 for Transformers, and stochastic gradient descent optimizer (SGD) for the discriminator head. In addition, we use a hyperbolic-tangent decay learning rate scheduler [35] during training for the transformer networks. Each network is trained with a maximum of 400 epochs; most experiments required less than 100 epochs. We also used early stopping to prevent overfitting, where training was halted when validation accuracy began to increase or became stagnant.

Performance compared to other methods. We run experiments comparing our method to various other methods for transfer learning, including domain adaptation and knowledge distillation. While each method serves different purposes, all learn from two different domains in the Sim2Real problem setting. Results for this experiment can be found in Table 1. In addition, because learn-to-steer has not yet been benchmarked for transformer networks, we show results of training each model from scratch in Table 2. An ablation study was also conducted to show the empirical effects on performance for each contribution in Table 3.

Complexity Tradeoff. In general, our results show similar generalization results to other methods for knowledge transfer, such as distillation and domain adaptation. While almost all methods showed improvement beyond the baseline ViViT model, there are stark differences in model complexity. Our contributions were able to achieve similar or better results on vision transformer architectures with little added complexity or walltime compared to the direct baseline training. A visualization of model complexity and walltime for each method in Table 1 can be found in Figure 3.

5 Conclusion and Discussion

Our work explores and improves the generalizability and performance of transfer learning for spatio-temporal Vision Transformers on the learn-to-steer task for autonomous driving. Although simple and direct pre-training achieves the highest performance in the target domain, it performs most poorly in unseen datasets in testing. Our method, which optimizes contrastive loss modified for behavioral cloning settings along with patch-wise augmentation, takes advantage of the temporal steering sequence to describe the semantic action being performed in a sample as well as properties of vision transformers. It is also simple to implement beyond standard pre-training compared to

Table 3: **Ablation results**. We show the effects of each contributions on performance compared to a baseline model trained on target data (Waymo). While contrastive learning seemed to benefit the ViViT model more, patch augmentation seemed benefit TimeSformer. Similarly to the previous table, the datasets which each model is trained on is denoted by "T" for "Target" and "S" for "Source".

Method	Datasets	Waymo*	Audi	SullyChen	Udacity	CARLA*
ViViT	T	88.8	75.9	88.8	50.0	66.2
ViViT+Contrastive	T	**91.9**	**79.8**	**96.6**	52.1	69.6
ViViT+PatchAug	T	89.9	76.2	83.6	47.6	67.8
ViViT+Ours	T	91.7	78.9	96.1	51.9	**70.5**
ViViT+Vanilla	S,T	92.2	79.2	94.8	**52.4**	69.8
TimeS	T	**93.8**	79.8	**93.5**	48.6	**68.6**
TimeS+Contrastive	T	93.1	79.1	91.0	50.2	67.8
TimeS+PatchAug	T	93.7	80.1	93.4	**51.7**	68.3
TimeS+Ours	T	**93.8**	79.3	91.8	51.0	67.9
TimeS+Vanilla	S,T	92.4	**80.3**	92.2	50.7	67.8

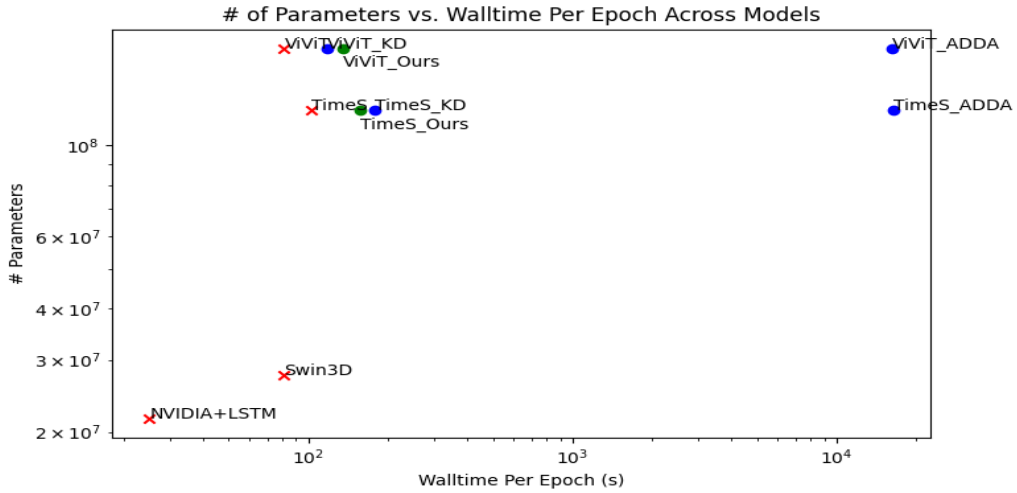

Figure 3: **Model size vs. Walltime plot.** We plot the number of trainable parameters in each model from Tables 1 and 2, and their respective training wall-time in seconds per epoch. Knowledge transfer methods are marked in blue, while baseline architectures are denoted with red X's. Our method is shown in green.

knowledge distillation and domain adaptation methods, which involve adding extra tokens or the training of an external discriminator network, and offers similar boosts in performance.

One downside of end-to-end steering is that the task is relatively new compared to others for autonomous driving. Additionally, many datasets currently available offer steering angles as an added bonus annotation, rather than being optimized for use with steering angle labels. Many datasets we used had a disproportionate distribution of steering angles, where over 90% of samples represented "straight" steering. Balancing these datasets drastically reduced the number of samples. To the best of our knowledge, datasets which prioritize diversity of steering actions do not exist yet, and would greatly benefit works for end-to-end steering. Another interesting path for future work could be using temporal control actions in multi-task settings, where control sequence labels can be used to characterize samples to improve performance of other tasks.

References

[1] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

[2] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo. Swin transformer v2: Scaling up capacity and resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[3] S. Yan, X. Xiong, A. Arnab, Z. Lu, M. Zhang, C. Sun, and C. Schmid. Multiview transformers for video recognition. Jan 2022. URL http://arxiv.org/abs/2201.04288.

[4] K. Li, Y. Wang, J. Zhang, P. Gao, G. Song, Y. Liu, H. Li, and Y. Qiao. Uniformer: Unifying convolution and self-attention for visual recognition, 2022.

[5] H. Akbari, L. Yuan, R. Qian, W.-H. Chuang, S.-F. Chang, Y. Cui, and B. Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *arXiv preprint arXiv:2104.11178*, 2021.

[6] C. Wu, Y. Li, K. Mangalam, H. Fan, B. Xiong, J. Malik, and C. Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. *Computer Vision and Pattern Recognition Conference (CVPR)*, 2022. URL https://arxiv.org/abs/2201.08383.

[7] Y. Zhang, X. Li, C. Liu, B. Shuai, Y. Zhu, B. Brattoli, H. Chen, I. Marsic, and J. Tighe. Vidtr: Video transformer without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13577–13587, 2021.

[8] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, page 6836–6846. openaccess.thecvf.com, 2021.

[9] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.

[10] A. Bulat, J.-M. Perez-Rua, S. Sudhakaran, B. Martinez, and G. Tzimiropoulos. Space-time mixing attention for video transformer. In *NeurIPS*, 2021.

[11] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[12] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[13] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[14] E. D. Cubuk, B. Zoph, J. Shlens, and Q. Le. Randaugment: Practical automated data augmentation with a reduced search space. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18613–18624. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d85b63ef0ccb114d0a3bb7b7d808028f-Paper.pdf.

[15] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[16] S. Yun, S. J. Oh, B. Heo, D. Han, and J. Kim. Videomix: Rethinking data augmentation for video classification. *arXiv preprint arXiv:2012.03457*, 2020.

[17] I. Budvytis, P. Sauer, T. Roddick, K. Breen, and R. Cipolla. Large scale labelled video data augmentation for semantic segmentation in driving scenarios. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[18] C. Sun, A. Nagrani, Y. Tian, and C. Schmid. Composable augmentation encoding for video representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8834–8844, October 2021.

[19] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[20] H. Kuang, Y. Zhu, Z. Zhang, X. Li, J. Tighe, S. Schwertfeger, C. Stachniss, and M. Li. Video contrastive learning with global context. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 3195–3204, October 2021.

[21] R. Li, Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei. Motion-focused contrastive learning of video representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2105–2114, October 2021.

[22] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1134–1141. IEEE, 2018.

[23] W. Zhao, J. P. Queralta, and T. Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, page 737–744, Dec 2020.

[24] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[26] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to end learning for self-driving cars. Apr 2016. URL http://arxiv.org/abs/1604.07316.

[27] Y. Shen, L. Zheng, M. Shu, W. Li, T. Goldstein, and M. Lin. Gradient-free adversarial training against image corruption for learning-based steering. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 26250–26263. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/dce8af15f064d1accb98887a21029b08-Paper.pdf.

[28] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator. In S. Levine, V. Vanhoucke, and K. Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, page 1–16. PMLR, 13–15 Nov 2017.

[29] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, and Others. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 2446–2454. openaccess.thecvf.com, 2020.

[30] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth. A2d2: Audi autonomous driving dataset. Apr 2020. URL http://arxiv.org/abs/2004.06320.

[31] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko. Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[32] S. Chen. A collection of labeled car driving datasets, https://github.com/sullychen/driving-datasets, 2018.

[33] Udacity. Udacity self-driving dataset, 2017. URL https://github.com/udacity/self-driving-car/tree/master/datasets.

[34] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, May 2019.

[35] B.-Y. Hsueh, W. Li, and I.-C. Wu. Stochastic gradient descent with hyperbolic-tangent decay on classification. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 435–442. IEEE, 2019.