Gaze-Informed Multi-Objective Imitation Learning from Human Demonstrations

Ritwik Bera, 1* Vinicius G. Goecks, 2* Gregory M. Gremillion, 2 Vernon J. Lawhern, 2 John Valasek, 1 Nicholas R. Waytowich 2,3

1 Texas A&M University, 2 U.S. Army Research Laboratory, 3 Columbia University
{ritwik, vinicius.goecks, valasek}@tamu.edu, {gregory.m.gremillion,vernon.j.lawhern.civ, nicholas.r.waytowich}.civ@mail.mil

Abstract
In the field of human-robot interaction, teaching learning agents from human demonstrations via supervised learning has been widely studied and successfully applied to multiple domains such as self-driving cars and robot manipulation. However, the majority of the work on learning from human demonstrations utilizes only behavioral information from the demonstrator, i.e. what actions were taken, and ignores other useful information. In particular, eye gaze information can give valuable insight towards where the demonstrator is allocating their visual attention, and leveraging such information has the potential to improve agent performance. Previous approaches have only studied the utilization of attention in simple, synchronous environments, limiting their applicability to real-world domains. This work proposes a novel imitation learning architecture to learn concurrently from human action demonstration and eye tracking data to solve tasks where human gaze information provides important context. The proposed method is applied to a visual navigation task, in which an unmanned quadrotor is trained to search for and navigate to a target vehicle in a real-world, photorealistic simulated environment. When compared to a baseline imitation learning architecture, results show that the proposed gaze augmented imitation learning model is able to learn policies that achieve significantly higher task completion rates, with more efficient paths, while simultaneously learning to predict human visual attention. This research aims to highlight the importance of multimodal learning of visual attention information from additional human input modalities and encourages the community to adopt them when training agents from human demonstrations to perform visuomotor tasks.

1 Introduction
In the field of human-robot interaction, learning from demonstrations (LfD), also referred to as imitation learning, is widely used to rapidly train artificial agents to mimic the demonstrator via supervised learning (Argall et al. 2009 Osa et al. 2018). LfD using human-generated data has been widely studied and successfully applied to multiple domains such as self-driving cars (Codevilla et al. 2019), robot manipulation (Rahmatizadeh et al. 2018), and navigation (Silver, Bagnell, and Stentz 2010). While LfD is a simple and straightforward approach for teaching intelligent behavior, it still suffers from sample complexity issues when learning a behavior policy directly from images, i.e. mapping what the robot’s camera sees to what action it should take. The majority of the work on LfD utilizes only behavioral information from the demonstrator, i.e. what actions were taken, and ignores other information such as the eye gaze of the demonstrator (Argall et al. 2009). Eye gaze is an especially useful signal, as it can give valuable insight towards where the demonstrator is allocating their visual attention (Doshi and Trivedi 2012), and leveraging such information has the potential to improve agent performance when using LfD.

Eye gaze is an unobtrusive input signal that can be easily collected with the help of widely available eye tracking hardware (Poole and Ball 2006). Eye gaze data comes at almost no additional cost when teleoperating robots to collect expert demonstration data, as the human operator is able to do the task naturally as before. Eye gaze acts as an important signal in guiding our actions and filtering relevant parts of the environment that we perceive (Schütz, Braun, and Gegenfurtner 2011), and as such, measuring eye gaze gives us an indication of visual attention that can be leveraged when training AI agents.

Previous works that have attempted to utilize eye-gaze for imitation learning (such as (Zhang et al. 2018)) have only done so in simple, synchronous environments, limiting their applicability to real-world domains. In this paper, we utilize...
multi-objective learning for gaze-informed imitation learning of real-world, asynchronous robotics tasks, such as autonomous quadrotor navigation, which have proven difficult for traditional imitation learning techniques (Ross, Gordon, and Bagnell 2011; Bojarski et al. 2016).

This work attempts to frame the problem of visual quadrotor navigation as a supervised multi-objective learning problem wherein our model has an input space comprised of onboard inertial measurements, RGB and depth images, and attempts to predict both the demonstrator’s actions as well as the the demonstrator’s eye gaze while performing a given task, as shown in Figure 1. The benefit behind using a multi-objective framework, as proposed by this work, enables eye gaze to act as a regularizer by introducing an inductive bias. In their study the authors highlight that gaze during subtask classification, reflecting the user internal reward function. In their study the authors highlight that gaze behavior for teaching could vary for more complex tasks that involve search and planning, such as the the one presented in this study.

Gaze-Augmented Imitation Learning Augmenting imitation learning with eye gaze data has been shown to improve policy performance when compared to unaugmented imitation learning (Zhang et al. 2018) and improve policy generalization (Liu et al. 2019) in visuomotor tasks, such as playing Atari games and simulated driving. [Xia et al. (2020) also proposed a periphery-fovea multi-resolution driving model to predict human attention and improve accuracy in a fixed driving dataset.

Attention Guided Imitation Learning (Zhang et al. 2018), or AGIL, collected human gaze and joystick demonstrations while playing Atari games and presented a two-step training procedure to incorporate gaze data in imitation learning, training first a gaze prediction network modeled as a human-like foveation system, then a policy network using the gaze predictions represented as attention heatmaps. The reported gains in policy performance vary from 3.4 to 1143.8%, depending on the game complexity and the number of sub-tasks the player has to attend to, which illustrates how the benefit of a multimodal learning approach is tied to the desired task to be solved. However, to eliminate effects of human reaction time and fatigue, eye-gaze was collected in a synchronous fashion in which the environment only advanced to the next state once the human took an action, and game time was limited to 15 minutes, followed by a 20 minutes rest period. This highlights the challenges of human-in-the-loop machine learning approaches and real-time human data collection. This limits the applicability of attention guided techniques to only relatively simple domains where the environment can easily be stopped and started to synchronously line up with human input, and will not work on real-world environments such as quadrotor navigation.

Liu et al. (2019) proposed two approaches to incorporate gaze into imitation learning: 1) using gaze information to modulate input images, as opposed to using gaze as an additional policy input; and 2) using gaze to control dropout rates at the convolution layers during training. Both approaches use a pre-trained gaze prediction network before incorporating it into imitation learning. Evaluated on a simulated driving task, Liu et al. (2019) showed that both approaches reduced generalization error when compared to imitation learning without gaze, with the second approach (using gaze to control dropout rates during training) yielding about 24% error reduction compared to 17% of the first approach when predicting steering wheel angles on unseen driving tasks.

3 Methods
We train a novel multi-objective model via supervised learning to predict actions to accomplish a given task and the location of eye-gaze of a human operating the quadrotor through a first-person camera view. The training is accomplished by collecting a set of human demonstrations where we record the observation space of the quadrotor (i.e. cameras and IMU sensors) as well as the demonstrator’s eye-gaze position and their input actions while performing a quadrotor search and navigation task.

Model Architecture
We take an end-to-end supervised learning approach to solving the visual search and navigation problem, aiming to avoid making any structural assumptions with regard to either the task being performed by the quadrotor, or the environment the quadrotor is expected to operate in. The model
Figure 2: Model architecture illustrating how quadrotor onboard sensor data and cameras frames are processed, how features are combined in a shared representation backbone, and how multiple outputs, i.e. the gaze and action prediction networks, are performed via independent model heads.

architecture is illustrated in Figure 2 and explained in the following paragraphs. The input space of the model is composed of three modalities:

- The quadrotor has access to a constant stream of RGB frames, 704 by 480 pixel resolution, captured by its onboard forward-facing camera, which are stored in an image buffer. Each frame is passed through a truncated version of a pre-trained ResNet34 (He et al. 2016), which includes only its first three convolutional blocks, generating a feature map of dimensions 128x28x28 (channel, height, width) that are stored in an intermediate buffer. Given the photorealistic scenes rendered by Microsoft AirSim’s Unreal Engine based simulator, ResNet was chosen as the pre-trained feature extractor as it was trained with real-world images. At each time instant \( t \), the feature map corresponding to time instant \( t \) is concatenated with the feature map corresponding to time instant \( t-8 \), across the channel dimension, generating a 256x28x28 dimensional tensor. The concatenated feature maps are passed through a sequence of trainable convolution layers, as described in Figure 2, forming an 484 dimensional vector of visual features and the first input to the model.

- The quadrotor also has access to depth images, which are generated by the simulated onboard camera from which the RGB frames are obtained and, consequently, sharing the same geometric reference frame. The depth images, 704 by 480 pixel resolution, are passed through a Histogram of Oriented Gradients (Dalal and Triggs 2005) feature extractor that generates a 512 dimensional feature vector. This forms the second input to the model, HOG Features in Figure 2.

- The quadrotor also outputs filtered data from its simulated gyroscope and accelerometer, providing the model access to the vehicle’s angular orientation, velocity, and acceleration, and linear velocity and acceleration, altitude, but no absolute position such as given by a GPS device, for a total of 16 features, as shown by Kinematic Features in Figure 2.

The resulting Visual, HOG, and Kinematic Features are concatenated, batch-normalized, and then passed through two fully connected layers with 128 output units each and ReLU activation function, forming the shared backbone of the proposed multi-headed model. The intermediate feature vector is then passed through two separate output heads, the Action Prediction Head and the Gaze Prediction Head, as seen in the right side of Figure 2. Each head has a hidden layer with 32 units and an output layer. The Action Prediction Head outputs 4 scalar values bounded between -1 and 1 representing joystick values that control the quadrotor linear velocities associated with movement in the x, y, and z directions as well as the yaw. The Gaze Prediction Head outputs 2 scalar values bounded between 0 and 1 representing normalized eye gaze coordinates in the quadrotor camera’s frame of reference.

Training Objective

The loss function used in the training routine is a linear combination of a Gaze Prediction Loss, \( L_{GP} \), and a Behavior Cloning Loss, \( L_{BC} \):

\[
L(\theta) = \lambda_1 \cdot \text{Gaze Prediction Loss}(L_{GP}) + \lambda_2 \cdot \text{Behavior Cloning Loss}(L_{BC}),
\]

where \( \theta \) represents the set of trainable model parameters and \( \lambda_1 \) and \( \lambda_2 \) are hyperparameters controlling the contribution of each loss component, defined as the mean-squared
error between ground truth and predicted values:

\[
\mathcal{L}_{GP} = \| (\pi_{gaze} - g') \|^2,
\]

\[
\mathcal{L}_{BC} = \| (\pi_{action} - a') \|^2,
\]

where \(g_t\) is a vector representing the true \(x, y\) coordinates of the eye gaze in the camera frame at instant \(t\), and \(a_t\) is the vector representing the expert joystick actions components for forward velocity, lateral velocity, yaw angular velocity, and throttle taken at that same time instant \(t\). \(\pi_{gaze}\) and \(\pi_{action}\) are the outputs of the Gaze Prediction and Action Prediction Heads, respectively.

**Task Description**

The experiments were conducted in simulated environments rendered by the Unreal Engine using the Microsoft AirSim (Shah et al. 2018) plugin as an interface to receive data and send commands to the quadrotor, a simulated Parrot AR.Drone vehicle. The task consisted of a search and navigate task where the drone must seek out a target vehicle and then navigate towards it in a photo-realistic cluttered forest simulation environment, as seen in Figure 3. This environment emulates sun glare and presents trees and uneven rocky terrain as obstacles for navigation and visual identification of the target vehicle.

**Data Collection Procedure**

The task was presented to the demonstrator on a 23.8 inches display, 1920x1080 pixel resolution, and eye gaze data collection was conducted using a screen-mounted eye tracker positioned at a distance of 61cm from the demonstrator’s eye. Before the data collection, the height of the demonstrator’s chair was adjusted in order to position their head at the optimal location for gaze tracking. The eye tracker sensor was calibrated according to a 8-point manufacturer-provided software calibration procedure. The demonstrator avoided moving the chair and minimized torso movements during data collection, while moving the eye naturally. The demonstrator was also given time to acclimate to the task until they judged for themselves that they were confident in performing it. To perform the task, the demonstrator was only given access to the first-person view of the quadrotor and no additional information about their current location or the target current location in the map. Using a Xbox One joystick, the demonstrator was able to control quadrotor throttle and yaw rate using the left joystick and forward and lateral velocity commands using the right stick, as is standard for aerial vehicles. The complete data collection setup is shown in Figure 5.

Training data is collected with both initial locations for the quadrotor and target vehicle randomly sampled without replacement from a set of 25 possible locations covering a pre-defined area of the map, as shown in Figure 4. This prevents invalid initial locations, such as inside the ground or trees, while still covering the desired task area. Initial quadrotor heading is also uniformly sampled from 0 to 360 degrees. In total, 200 human demonstration trajectories representing unique quadrotor and vehicle location pairs (from a total number of \(25 \times 24 = 600\) pairs) were collected to form an expert dataset for model training and testing. This includes RGB and depth images, the quadrotor’s inertial sensor readings, and eye gaze coordinates.

**Experimental Setup**

The entire dataset of 200 trajectories is split into a training set of 180 trajectories and a test set of 20 trajectories, for each experimental run. During exploratory data analysis on the collected dataset, it was found that the yaw-axis actions were mostly centered around zero with a very low variance. To ensure that the agent is trained with enough data on how to properly turn and not just fly straight, weighted oversampling was employed in the training routine.

Evaluation of each trained model is performed by conducting rollouts in the environment as follows. A configuration is defined as a pair consisting of an initial spawn
In terms of the hardware infrastructure, all experiments were run in a single machine running Ubuntu 18.04 LTS OS equipped with an Intel Core i9-7900X CPU, 128 Gb of RAM, and NVIDIA RTX 2080 Ti used only for Microsoft AirSim. In terms of the software infrastructure, the proposed model was implemented in Python v3.6.9 and PyTorch (Paszke et al. 2019) v1.5.0, data handling using Numpy (Van Der Walt, Colbert, and Varoquaux 2011) v1.18.4 and Pandas (Wes McKinney 2010) v1.0.4, data storage using HDF5 via h5py v2.10.0, data versioning control via DVC v1.6.0, and experiment monitoring using MLflow (Zaharia et al. 2018) v1.10.0. Random seeds were set in Python, PyTorch, and NumPy using the methods random.seed(seed_value), torch.manual_seed(seed_value), and numpy.random.seed(seed_value), respectively. With respect to hyperparameters, the loss function is optimized using the Adam (Kingma and Ba 2014) optimizer with a learning rate of 0.0003 for 20 epochs and batch size of 128 data samples. Hyperparameters for model size, such as number of layers and units, are described in the “Model Architecture” section above. Hyperparameter tuning was done manually and changes in network size had minor impact in the proposed approach performance.

**Metrics**

In real-world robotics based applications such as the one pursued in this work, standard machine learning metrics such as validation losses, and others that quantify how well a model has been trained with the given data, tend to not translate into actual high performance models that can successfully satisfy end-user requirements. To that effect, this research employs certain custom metrics to evaluate the performance of our model when deployed to a new test scenario and help demonstrate the effectiveness of using eye gaze as an additional supervision modality. The core metrics for comparison between the proposed model and baselines include:

- **Task Completion Rate**: this is evaluated by dividing the number of successful completions of the task by the total number of rollouts performed with that model. A successful task completion is defined as the quadrotor finding the target vehicle and intercepting it within a 5 meter radius.

- **Collision Rate**: this is evaluated by dividing the number of collisions during a rollout with the total number of rollouts performed with that model.

- **Success weighted by (normalized inverse) Path Length (SPL)**: as defined in (Anderson et al. 2018), is evaluated for episodes with successful completion by dividing the shortest-path distance from the agent’s starting position to the target location with the distance actually taken by the agent to reach the goal. In this work, it is assumed the shortest-path distance is a straight line from the initial quadrotor location to the target vehicle location, irrespective of obstacles.

### 4 Results

The proposed gaze-augmented imitation learning model, denoted Gaze BC in this section, is compared against a standard imitation learning model with no gaze information, denoted Vanilla BC in this section. Note that, when training our proposed gaze-augmented imitation learning model, the values of both $\lambda_1$ and $\lambda_2$ in Equation 1 are set at 1.0, while for training the baseline imitation learning model with no gaze information, $\lambda_1$ is set to 0.0 while $\lambda_2$ stays at 1.0.

In terms of policy performance, Table 1 provides a summary of the relevant metrics. Task Completion Rate can be seen to be higher for the Gaze BC model by approximately 13.7 percentage points. A Wilcoxon signed-rank test, a non-parametric statistical hypothesis test, was performed using
the task completion rate values (one metric value per seed) for both models, to evaluate the statistical significance of the obtained results. A p-value of 0.0312 was obtained which shows that our Gaze BC model outperforms the Vanilla BC model in statistically significant terms.

Collision rate remains roughly the same for both the Gaze BC and the Vanilla BC model. This can be attributed to the fact that both models have access to the same observation space (visual features, HOG features and kinematic features) and thus both models have the information needed for collision avoidance.

SPL for the Gaze BC model can be seen to be 1.75x the SPL for the Vanilla BC model. This shows that whenever the models successfully accomplish the task, the Gaze BC model, on average, takes a significantly shorter path to reach the goal than the Vanilla BC model. This result again illustrates the superior performance of a policy trained with eye-gaze data.

5 Discussion
In this work we proposed a novel imitation learning architecture to learn concurrently from human actions and eye gaze to solve tasks where gaze information provides important context. Specifically, we applied the proposed method to a visual search and navigation task, in which an unmanned quadrotor is trained to search for and navigate to a target vehicle in an asynchronous, photorealistic environment. When compared to a baseline imitation learning architecture, results show that the proposed gaze augmented imitation learning model is able to learn policies that achieve significantly higher task completion rates, with more efficient paths while simultaneously learning to predict human visual attention.

The closest related work to ours is Attention Guided Imitation Learning (Zhang et al., 2018), or AGIL, which presented a two-step training process to train a gaze prediction network then a policy to perform imitation learning on Atari games, as explained in the Related Work section. We did not perform a direct comparison to AGIL due to fact that AGIL currently operates on synchronous environments which can be paused, allowing for the game states to be synchronized with the human data. The data collection process for AGIL requires the advancing of game states one step at a time after the human performs each action. This is feasible when working with simple environments such as Atari where the game engine can be paused, however we aimed to develop an approach that could be translated to real-world applications running asynchronously in real time. Moreover, AGIL’s architecture does not include any pretrained feature extractor, which significantly increases data collection requirements when dealing with vision-based robotics tasks like the one featured in this work.

With respect to the gaze prediction performance, four distinct gaze patterns observed during the human demonstrations were also learned by the Gaze Prediction Head of the proposed model, as seen in Figure 6: 1) “motion leading” gaze pattern where gaze attends to the sides of the images followed by yaw motion in the same direction, as illustrated in Figure 6a. This pattern is mostly observed at the beginning of the episode when the target vehicle is not in the field-of-view of the agent; 2) “target fixation” gaze pattern where gaze is fixed at target during the final approach, as illustrated in Figure 6b. In this pattern gaze is fixed at the top of the target vehicle, independent of the current motion of the agent; 3) “saccade” gaze pattern where gaze rapidly switches fixation between nearby obstacles (Salvucci and Goldberg 2000), as illustrated in Figure 6c. This pattern is characteristic when there are multiple obstacles between the current agent location and the target vehicle; and 4) “obstacle fixation” gaze pattern where gaze attends to nearby obstacles when navigating to the target, as learned from multiple demonstrations, which might differ from the current obstacle user is attending to at the moment, as illustrated in Figure 6d. This pattern is mostly observed when the quadrotor is in close distance to an potential obstacle. This illustrates how the Gaze Prediction Head of the proposed model was able to capture and replicate similar visual attention demonstrated by the user when performing the task.

### Limitations and Future Work
A limitation of the current work is not being able to completely the proposed task with a higher rate. We argue that one simple reason for this issue, as in all end-to-end learning approaches, is the limited training dataset. This issue is accentuated in human-in-the-loop machine learning tasks, such as the one used in this work, since human-generated data is scarce. For this work we only had available 200 human-generated trajectories and believe that task completion rate could increase by simply training the model with more data. This additional data could come from more independent trajectories, by labelling agent-generated trajectory and aggregating to the original dataset (Ross, Gordon, and Bagnell 2011), or by tasking the human to oversee the agent and perform interventions when the agent’s policy is close to fail, also aggregating this intervention data to the original dataset (Goecks et al. 2019). In this case, the ability of a human trainer to enact timely interventions may be greatly improved by the gaze predictions the model generates, which would highlight instances where this artificial gaze deviates from what they expect, potentially indicating subsequent undesirable behavior and providing some level of model interpretability or explainability.

Another possible avenue for future work is the addition of a one-hot encoded task specifier in the input space of the model, which could aid in training models that learn,
like humans, to adapt their attention and behavior according to the task at hand. Furthermore, the integration of additional human input modalities to the proposed approach, such as natural language, could similarly be used to condition the model to perform multiple different tasks. The combination of gaze and natural language would enable humans to more naturally interact with learning agents, and enable those agents to disambiguate demonstrator behavior and attention dynamics that are otherwise ambiguous because they are pursuant to distinct tasks and goals that could be grounded to language commands. The ability to use eye-gaze data to leverage a human’s visual attention opens the door to adapting this research to learning unified policies that can generalize across multiple context- and goal-dependent, tasks.

Acknowledgments

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-20-2-0114. 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 Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

6 Ethics Statement

Imitation learning-based agents, to a greater degree when learning from human demonstrations, are susceptible to learning the bias present in the training dataset. Specifically related to this work, the learning agent would learn to replicate action and gaze patterns demonstrated in the training dataset. This issue can best be addressed during the data collection phase, where action and gaze information would be collected from a diverse pool of users with different background that could influence their action and gaze patterns, and consequently, the task execution. A positive ethical impact of learning from additional input modalities is being able to capture more information from humans, which in turn can be used to better train artificial intelligent agents that can model and act according to human preferences.

References

Abu-Mostafa, Y. S. 1990. Learning from hints in neural networks. *Journal of complexity* 6(2): 192–198.

Anderson, P.; Chang, A. X.; Chaplot, D. S.; Dosovitskiy, A.; Gupta, S.; Kolotun, V.; Kosecka, J.; Malik, J.; Mottaghi, R.; Savva, M.; and Zamir, A. R. 2018. On Evaluation of Embodied Navigation Agents. *CoRR* abs/1807.06757. URL http://arxiv.org/abs/1807.06757.

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57(5): 469–483.

Baxter, J. 2000. A model of inductive bias learning. *Journal of artificial intelligence research* 12: 149–198.

Bojarski, M.; Testa, D. D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L. D.; Monfort, M.; Muller, U.; Zhang, J.; Zhang, X.; Zhao, J.; and Zieba, K. 2016. End to End Learning for Self-Driving Cars. *CoRR* abs/1604.07316. URL http://arxiv.org/abs/1604.07316.

Caruana, R. 1993. Multitask Learning: A Knowledge-Based Source of Inductive Bias. In *Proceedings of the Tenth International Conference on International Conference on Machine Learning*, ICML’93, 41–48. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN 1558603077. doi: 10.5555/3091529.3091535.

Codelvilla, F.; Santana, E.; López, A. M.; and Gaidon, A. 2019. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, 9329–9338.

Dalal, N.; and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE computer soci-
Doshi, A.; and Trivedi, M. M. 2012. Head and eye gaze dynamics during visual attention shifts in complex environments. *Journal of vision* 12(2): 9–9.

Goecks, V. G.; Gremillion, G. M.; Lawhern, V. J.; Valasek, J.; and Waytowich, N. R. 2019. Efficiently Combining Human Demonstrations and Interventions for Safe Training of Autonomous Systems in Real-Time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2462–2470. AAAI Press. doi:10.1609/aaai.v33i01.33012462.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Liu, C.; Chen, Y.; Tai, L.; Liu, M.; and Shi, B. E. 2019. Utilizing Eye Gaze to Enhance the Generalization of Imitation Networks to Unseen Environments. *CoRR* abs/1907.0. URL http://arxiv.org/abs/1907.04728.

Osa, T.; Pajarinen, J.; Neumann, G.; Bagnell, J. A.; Abbeel, P.; and Peters, J. 2018. An Algorithmic Perspective on Imitation Learning. *Foundations and Trends® in Robotics Systems* 3(1-2): 1–179. ISSN 1935-8253. doi:10.1561/2300000053. URL http://dx.doi.org/10.1561/2300000053.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d’Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Poole, A.; and Ball, L. J. 2006. Eye tracking in HCI and usability research. In *Encyclopedia of human computer interaction*, 211–219. IGI Global.

Rahmatizadeh, A.; Abolghasemi, P.; Bölöni, L.; and Levine, S. 2018. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 3758–3765. IEEE.

Ross, S.; Gordon, G.; and Bagnell, D. 2011. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In Gordon, G.; Dunson, D.; and Dudík, M., eds., *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, 627–635. Fort Lauderdale, FL, USA: PMLR. URL http://proceedings.mlr.press/v15/ross11a.html.

Ruder, S. 2017. An Overview of Multi-Task Learning in Deep Neural Networks.