2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Neural Network Controller for Autonomous Pile Loading Revised

Wenyan Yang[1], Nataliya Strokina[1], Nikolay Serbenyuk[2], Joni Pajarinen[3,4],
Reza Ghabcheloo[2], Juho Vihonen[5], Mohammad M. Aref[5] and Joni-Kristian Kämäräinen[1]

***Abstract*— We have recently proposed two pile loading controllers that learn from human demonstrations: a neural network (NNet) [1] and a random forest (RF) controller [2]. In the field experiments the RF controller obtained clearly better success rates. In this work, the previous findings are drastically revised by experimenting summer time trained controllers in winter conditions. The winter experiments revealed a need for additional sensors, more training data, and a controller that can take advantage of these. Therefore, we propose a revised neural controller (NNetV2) which has a more expressive structure and uses a neural attention mechanism to focus on important parts of the sensor and control signals. Using the same data and sensors to train and test the three controllers, NNetV2 achieves better robustness against drastically changing conditions and superior success rate. To the best of our knowledge, this is the first work testing a learning-based controller for a heavy-duty machine in drastically varying outdoor conditions and delivering high success rate in winter, being trained in summer.**

I. INTRODUCTION

Pile loading is one of the most challenging tasks in earth moving automation for Heavy-duty mobile (HDM) machines. This is partly caused by the difficulty of modelling the interaction between the tool and the material [3] and partly because of high variation in work sites and weather conditions throughout the year (Fig. 1). Weather conditions affect the material properties, the hydraulics properties of the machine, and the ground surface properties. The state-of-the-art works on pile loading or excavation automation are either model-based or use heuristics [4], [5], and experimented only in simulators or with toy setups. Therefore it is unclear how well these methods perform in real work sites.

Recently, there have been attempts to learn a controller from human demonstrations using machine learning techniques to approximate the controller. Neural Network (NN) based controllers are proposed in Dadlich et al. [6] and Halbach et al. [1] and both were tested with real wheel loaders. Dadlich et al. concluded that *"different networks are needed to be trained with data collected in different conditions"*. That was verified in the experiments of Yang et al. [2] who showed that the NNet controller by Halbach et al. fails when test conditions are changed (e.g. distance to the pile). As a solution Yang et al. propose a Random Forest (RF) based controller that achieves clearly better success rate than NNet. Although RF performs well in the pile loading task, neural network controllers have beneficial properties for machine learning in robotics. In particular, neural networks are differentiable making them suitable for autonomous learning and exploration using the popular policy gradient Reinforcement Learning techniques [7], [5], [8].

Fig. 1: The training demonstrations were recorded during summer time (previously used in [2]). The wheel loader test experiments were conducted during winter time in various conditions over 30 days: cold wet weather (A), mild icy surface and partial snow (B), icy surface and snow covering the surface and pile (C), cold winter conditions and gravel moving partly replaced with snow clearing (D).

In this work, we propose neural network controllers (ANNet, DANNet) that are competitive against the prior arts. The attention module selects important signals during the different states of the pile loading task. In the winter experiments, these changes improve the performance of NNet from 0% to 100%. In addition, our previous results and findings are drastically revised thanks to the experiments conducted in winter with controllers trained using summer demonstrations. Our work provides the following contributions:

- A neural network controller with a neural attention module, which selects sensors that are important at different states of the pile loading control problem. It prevents neural controller's failure in changing environmental and dynamic load conditions.
- We highlight the loss of useful information in the cases of downsampling or "filtering" suitable input data (manual selection of the best demonstrations as done in the previous works), which have negative effect on the success rate.
- We demonstrate that the previous works lack controller state observability by experimentally comparing a variety of available sensors. Particularly in winter,

[1]Computing Sciences, Tampere University, Finland, [2]Automation Technology and Mechanical Engineering, Tampere University, [3]Department of Electrical Engineering and Automation, Aalto University, [4] Intelligent Autonomous Systems, Technische Universität Darmstadt, Germany and [5] Cargotec Oyj

the previously omitted hydraulic pressure at telescope joints plays a significant role in observability of load dynamics.

All controllers were implemented and experimented on a real-scale robotic wheel-loader. Testing was performed on multiple winter days over a 30 day period of time at different locations. Over this time the weather conditions changed dramatically, including an icy road, frozen material, wet snow and mud (see Fig. 1). This allowed us to verify the findings in highly diverse test conditions. The code and dataset will be published.

II. RELATED WORK

Autonomous pile loading works adopt heuristics [9] or are model-based [10], and are experimented only in a simulator [9] or toy-scale setups [11], [4], which cannot capture the complicated phenomena of the real-world problem. Jud et al. [11] utilize the trajectory of end-effector force-torque instead of the end-effector position to learn autonomous excavation. This way the model avoids generating arbitrarily high forces. In [9] Fernando et al. present a heuristic algorithm to learn an admittance controller for autonomous loading from a muck pile with non-homogeneous material. The proposed algorithm learns to apply specific forces instead of learning the desired trajectory. Sotiripoulos and Asada [10] use the power transmitted from the excavator to the soil as an input for adaptive excavation algorithm. By maximizing the output product of force and velocity the method enables bucket filling control. In the follow-up work on a similar set-up, [4] presents a feedback controller for rock scooping that optimizes a cost function using a Gaussian Process model to predict rock behaviour.

Model-based approaches succeed in many robotics applications. However, in pile loading the interaction between the bucket and the material is hard to model accurately. Several works attempt to learn this interaction using *learning from demonstrations*. Dadhich et al. [3] fit linear regression models to the lift and tilt bucket commands recorded with a joystick. Fukui et al. [12] use a neural network model that selects a pre-programmed excavation motion from a dataset of motions. [6], [1], [2] report real experiments of autonomous scooping with a real-scale HDM machine. Dadlich et al. [6] propose a shallow time-delay neural network controller. The controller uses the joint angles and velocities as inputs. After outdoor experiments the authors conclude that for different conditions the network controller needs to be retrained. Halbach et al. [1] train a shallow neural network controller (NNet) for bucket loading based on the joint angles and hydraulic drive transmission pressure. Yang et al. [2], presented a RF pile loading controller trained using a few demonstrations and the same sensors as Halbach et al. In field experiments RF clearly outperformed the NNet controller. In this work we revise the findings of Halbach and Yang et al. that are invalid if the training and test conditions are drastically different.

III. METHODS

In *learning from demonstrations* or *imitation learning* a controller is learned from human demonstrations. A number of sensor readings $\mathbf{s}_i$ are observed at each discrete time step i. For timestep i, The controller takes the sensor readings $\mathbf{s}_i$ as input and outputs control actions $\mathbf{u}_i$ that approximate the human actions The observation-action pairs constitute the training set D of demonstrations: $D = \{\langle \vec{s}_i, \vec{u}_i \rangle\}_{i=1..T}$, where T is the total number of samples. This yields a supervised learning problem, where the control actions are predicted by a function approximator F with the parameters $\vec{\Theta}$:

$$\vec{u} = F(\vec{s}; \vec{\Theta}) \ . \tag{1}$$

The approximator function is optimized to fit to the expert demonstrations using a suitable loss function ℓ:

$$\min_{\mathbf{\Theta}} \sum_{i=1}^{T} \ell \left(F(\mathbf{s}_i; \mathbf{\Theta}), \mathbf{u}_i \right) \ . \tag{2}$$

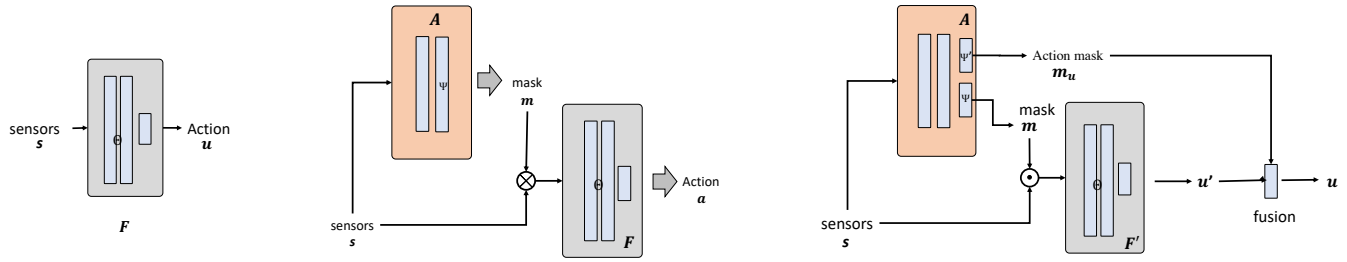
The standard loss is the Mean Squared Error (MSE)

$$\ell \left(F(\mathbf{s}_i; \mathbf{\Theta}), \mathbf{u}_i \right) = \frac{1}{T} \sum_{i=1}^{T} ||F(\mathbf{s}_i; \mathbf{\Theta}) - \vec{u}_i||_2^2 \ . \tag{3}$$

NNetV2 – A popular choice for the approximator F is a neural network. Hallbach et al. [1] propose a shallow fully connected conventional Multi-layer Perceptron (MLP) regressor network [13]. Their network has only five neurons in a single layer ($\vec{s}$-5-$\vec{u}$) and it is trained using the Levenberg-Marquardt (L-M) backpropagation. A small MLP trained has very limited expression power to effectively represent complex control policies (verified in our simulations). We revised the NNet network to NNetV2 which has three orders of magnitude more weights (Fig. 2(a)). NNetV2 has 200 neurons on two full-connected layers ($\vec{s}$-200-200-10-$\vec{u}$). The number of the inputs $\vec{s}$ and outputs $\vec{u}$ are the same in NNet and NNetV2 and they correspond to the HDM sensor and control signals.

Neural attention – Neural attention has many successful applications, for example, in computer vision [14], [15], natural language processing [16] and robotics [17], [18]. The main function of the "attention module" is to strengthen features important for the target task and suppress features that are less important [15]. For unseen test samples attention helps to attenuate noise produced by redundant sensors.

The purpose of applying attention module in this work was driven by following motivations: (1) attention module shall automatically select important features for corresponding actions, which improves robustness of the neural network controller against a changing environment and conditions; (2) the attention module is able to make the black box controller more explainable.



(a) NNetV2 (b) Single attention architecture (ANNet) (c) Dual attention (DANNet)

Fig. 2: The three proposed neural network controller architectures experimented in this work.

A. Neural attention module design

The attention mechanism is implemented as a fully connected neural network structure ($\vec{s}$-64-64-$\vec{m}$) that takes the sensor signals as input and applies an attention mask $\vec{m}$ to the same inputs before they are given to the NNetV2 controller (Fig. 2(b)). The attention controller is denoted as "ANNet" in our experiments. As a novel solution we also experiment with "dual attention" (DANNet in Fig. 2(c)) that provides attention masks for both the inputs and outputs.

1) Attention Neural Network Controller (ANNet): The architecture of the proposed attention neural network controller is illustrated in Fig. 2(b). It is a full-connected network with ReLU and Dropout layers: $\vec{s}$-64-64-$\vec{m}$, where $\vec{s}$ and $\vec{m}$ refer to the input and mask vectors.

The attention network A takes the sensor signals $\vec{s}$ as the input and outputs an "attention feature vector" $\vec{f}$ as

$$\vec{f} = A(\vec{s}; \vec{\Psi}) \ , \tag{4}$$

where $\vec{\Psi}$ are the attention network parameters. In the experiments, we investigate whether additional sensors improve attention and in that case the input with additional sensors is denoted as $\vec{s}'$.

The attention features are normalized by the softmax operation producing an *attention mask* $\vec{m}$:

$$m_j = softmax(f_j) = \frac{exp(f_j)}{\sum_{i=1}^{N} exp(f_i)} \tag{5}$$

or more compactly written as:

$$\mathbf{m} = softmax\left(A(\vec{s}; \mathbf{\Psi})\right) \ . \tag{6}$$

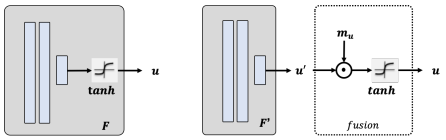
Finally, the original controller approximator equation in (1) is modified using dot product with the attention mask:

$$\mathbf{u} = F(\mathbf{s} \cdot \mathbf{m}; \mathbf{\Theta}) \ . \tag{7}$$

The attention neural network controller ANNet is optimized using the following optimization problem

$$\min_{\vec{\Theta}, \vec{\Psi}} \sum_{i=1}^{T} \ell\left(\vec{s}_i \cdot F(A(\vec{s}_i; \mathbf{\Psi}); \mathbf{\Theta}), \vec{u}_i\right) \ . \tag{8}$$

ANNet is trained using the same MSE loss (Eq. 2) as NNetV2.



(a) NNet and ANNet controller F (b) DANNet controller F' with action mask fusion

Fig. 3: The outputs of the both NNet and ANNet are generated by the $\tanh$ nonlinearity (a). The main difference is that in the DANNet the output is first modulated (dot product) by the output attention vector $\vec{m}_u$ (b).

2) Dual Attention Neural Network Controller (DANNet): For the DANNet architecture we introduce an additional attention mask $\vec{m}_u$ for the controller output. The control attention mask defines which control signals are "active". The intuition behind this idea comes from the demonstrations data itself - human drivers rarely perform more than one action at the same time. In 88% of the observation-action pairs in all recorded sequences there is only one action active. The output action mask is generated from the same sensor signals $\vec{s}$ as the input mask. Therefore, the same attention network A is used, but its output is augmented to produce an output attention feature vector of the size of the control signal $\vec{u}$ as $\vec{s}$-64-64-$\langle \vec{m}, \vec{m}_u \rangle$ where

$$\mathbf{m}_u = softmax\left(A(\mathbf{s}; \mathbf{\Psi}')\right) \ , \tag{9}$$

$|\mathbf{m_u}| = |\mathbf{u}|$ and $\vec{\Psi}'$ denotes the attention network parameters trained for dual attention mechanism.

As a small difference for the DANNet as compared to NNet and ANNet we add the control signal mask $\vec{m}_u$ inside the last nonlinearity function $\tanh$ as

$$\mathbf{u} = \tanh(\mathbf{m_u} \cdot \mathbf{u}') \tag{10}$$

where $\vec{u}'$ is the F controller output before the nonlinearity (Fig. 3). Otherwise, the loss function and training procedures are equivalent to NNetV2 and ANNet.

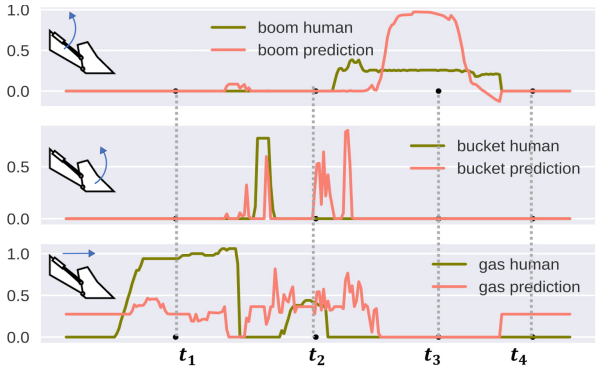
B. Training details

The NNetV2 controller structure is -200-200-10- and the attention module structure of ANNet and DANNet is -64-64-. The two last hidden layers are regulated by setting their dropout probabilities to 0.35 and ReLU is used after each hidden layer. The output layer of the controller network F

has three units with tanh-activation and it thus produces output control signal vector $\mathbf{u} \in [-1.0, +1.0]_{3\times1}$ that corresponds to normalized delta velocities (increment/decrease) of the control variables.

The attention module A uses the same sensor signals $\vec{s}$ except in the extra experiments where the attention module is augmented with additional sensors (see Table I). Output control (action) vector $\vec{u}$ for all models is a three-dimensional vector: $\vec{u} = \langle u_{\theta_1}, u_{\theta_2}, u_g \rangle$ that denotes the delta control variables (velocities). All models were trained using the RAdam (Rectified Adam) optimizer [19] with the mini-batch size 512 and the initial learning rate set to 0.001. The network weights were initialized using the Kaiming initialization [20]. All training converged after 150 epochs (see Section IV-F for more details).

Fig. 4(a) demonstrates pile loading action control signals for human and DANNet and Fig. 4(b) shows the attention values for different control signals at the time stamps t_1-t_4. At t_1, the main sensor attention is on the driving pressure p_d and action attention on the gas command u_g (approach pile). At time stamp t_2 and t_3, the sensor attention has moved to all input sensors while the action attention prefers the bucket u_{θ_2} and boom rising u_{θ_1} actions (raise the boom).



(*a*) One scooping demonstration and learned prediction

	Sensor soft mask				Action soft mask		
t_1	0.002	0.000	0.998	0.000	0.000	0.000	1.000
t_2	0.298	0.175	0.391	0.136	0.000	0.386	0.614
t_3	0.295	0.191	0.384	0.131	1.000	0.000	0.000
t_4	0.001	0.000	0.999	0.000	0.000	0.000	1.000
	θ_1	θ_2	p_d	p_t	u_{θ_1}	u_{θ_2}	u_g

(*b*) Learned attention masks

Fig. 4: Validation of the input (sensor) and output (control) masks. (a) human and DANNet control signals during a single pile loading action and (b) input and output attention masks at the four time stamps: t_1 (drive toward the pile), t_2 (impact with the pile), t_3 (bucket filling with boom) and t_4 (finished).

IV. EXPERIMENTS

Overview of the experiments – The robot hardware setup is described in Sec. IV-A. The experimental setup is introduced in Sec. IV-B. The first experiment (Experiment 1) compares the three controllers trained with the original summer data and tested in winter conditions in Sec. IV-C. In Experiment 2 the winter experiments are repeated using the full original dataset (no down-scaling and no manual selection) and with the additional sensors to improve observability (Sec. IV-D). Further ablation study on the neural attention modules is presented in Sec. IV-E with real data and more detailed analysis with offline validation data in Sec. IV-F.

A. Robot hardware

Wheel loader – The scooping (bucket filling) experiments were conducted on Avant 635, a commercial wheel loader with articulated-frame steering, except its hydrostatic transmission power with by-wire controllers. It carries a three degrees-of-freedom manipulator in the vertical plan consisting of a boom, telescope, and bucket joints collinear with each other. Its bucket is positioned in vertical plane by two joints, the boom joint and the bucket joint, and in horizontal plane by drive (throttle/gas) and articulated by a frame steering mechanism.

Sensors – The sensor and control signals used in this work are summarized in Table I. The main difference to the previous works [1], [2] is that there is an additional pressure sensor, *hydraulic pressure at the telescope joint*, which importance for robustness we experimentally demonstrate.

TABLE I: Avant wheel loader sensor and control signals (the necessary sensor for winter conditions is highlighted).

Sensor signals $\vec{s}$	
θ_1	Boom joint angle
θ_2	Bucket joint angle
p_d	Hydraulic drive transmission pressure
p_t	Hydraulic pressure at the telescope joint
Additional attention signals $\vec{s}'$	
p_l	Hydraulic pressure at the boom joint
p_b	Hydraulic pressure at the bucket joint
a	HST pump angle, proportional to driving speed
Control signals $\vec{u}$	
u_{θ_1}	Boom joint control
u_{θ_2}	Bucket joint control
u_g	Throttle (gas) command

Controller hardware – The control system is composed of multiple layers. In the actuator level and on control area network (CAN), industrial micro-controllers implement the power management and safety functions. In the PC control level, a Simulink Realtime target executes time-critical modules such as localization. Sub-systems communicate sensor data and control commands via UDP running on a Jetson AGX Xavier (8-Core ARM v8.2 64-bit NVIDIA Carmel CPU and 512-core NVIDIA Volta GPU with 64 Tensor Cores) on-board. The data collection, and closed-loop control are implemented on Jetson. Learning is performed offline on a standalone machine. Sensor data was received by a separate UDP thread at 20Hz rate. Overall system performance was about 8Hz, but it was reduced to 3Hz to make execution of commands more feasible for Avant.

B. Experimental setup

Bucket filling task – The experiments were conducted at an outdoor test site. The human demonstrations (training data) were the same as in our previous work [2] and therefore all results are comparable. All test experiments were conducted in the period of one month in winter conditions (Fig. 1). During the experiments the test site ground was frozen, muddy, slippery or dry varying on each day. The material properties changed as well, for example, partially frozen gravel, moist gravel and wet snow. The wheel loader performed the task learnt by the controller - drive up to the pile (varying distance and angle) and perform a scoop.

Performance measure – The bucket load after each test run was manually classified to be either successful scoop or not. A successful scoop was recorded when the bucket was at least half-full and otherwise a failure was marked. For all experiments, we report the success rate, $\frac{N_{full}}{N} \cdot 100\%$ as the performance indicator. The test runs were conducted on multiple different days over a period of 30 days in different weather conditions. The distance to the pile was varied between one to five meters and the wheel loader was positioned approximately toward the pile.

Training data – Training data consists of the 72 demonstrations from [2] collected during the summer of 2019, where 52 of the demonstration finish with full bucket and are therefore "ideal demonstrations". The low-level sensor measurements were down-sampled to synchronize them with the video input (20 Hz). Using the same data we define two different training sets:

- D_I: All 72 recorded human demonstrations using the original 500Hz sampling frequency for all sensor signals (total of 709,368 samples).
- D_{II}: Manually selected and temporally down-sampled (20Hz) data based on D_I (52 best demonstrations all finishing the bucket full). Total of 16,322 samples (observation-action pairs) are available.

Controller structures – NNet [1] has one hidden layer with 5 neurons. NNetV2 is described in Sec. III. RF controller [2] has 20 random trees and the maximum depth is 30.

C. Experiment 1: Transfer from summer to winter

As can be seen from Table II, both NNet, NNetV2 and RF controllers failed in the winter experiments using the data collected in summer for the training. The main failure cases were early boom rising and no boom rising at all.The results indicate that with such a big change in conditions the controllers trained by supervised learning simply cannot generalize. The results indicate both simple MLP-like neural network and RF controllers cannot be generalized.

D. Experiment 2: Adding data and telescopic joint pressure

In this experiment, we consider the dataset D_{II} as the baseline dataset. We first added more data simply by using all recorded demonstrations to create a new dataset D_I:

- including the ones with the half full bucket

TABLE II: Success rates of the controllers. Winter experiments consist of 15 test runs for each controller and are conducted over the time period of one month. Inputs of the controllers are $< \theta_1, \theta_2, p_d >$.

	Train: summer dataset D_{II}		
	NNet [1]	RF [2]	NNetV2
Test: summer	0%	80%	12%
Test: winter	0%	40%	0%

- keeping the original data recorded at the rate of 500Hz.

Except expanding the amount of training pairs, we also added one extra input sensor dimension p_t to the input sensors $\vec{s} = \langle \theta_1, \theta_2, p_d, p_t \rangle$ used in the previous work $\vec{s} = \langle \theta_1, \theta_2, p_d \rangle$.

TABLE III: Success rates using all data and with and w/o the additional pressure sensor p_t (hydraulic telescopic pressure). Approximately 30 attempts were executed with each controller over the period of 30 days.

Controller	p_t	Training data (summer) D_{II} [2]	D_I
RF		40%	30%
RF	✓	86%	87%
NNet		0%	0%
NNet	✓	0%	0%
NNetV2		0%	56%
NNetV2	✓	0%	76%

Effect of expanded dataset D_I – The bigger neural network NNetV2 controller obtained a clear improvement from 0% to 56% with D_I. The simple NNet[1] was not either improved by additinal sensor dimension nor the expanded dataset.

Effect of additional sensor p_t – With the additional sensor, p_t, NNetV2 was improved from 56% to 76% and RF from 30% to 87%. Clearly the hydraulic telescopic pressure sensor provides important information that makes the unknown controller states more observable. Our finding is that the original drive transmission pressure p_d is affected by wheel slip on icy surface while the telescopic joint pressure remains unaffected and correctly triggers boom rise.

E. Experiment 3: Neural attention

In Section III-A we introduce ANNet that learns to mask the input signals with the attention mask and DANNet that masks both the input sensor and output control signals. The results for all proposed neural network controllers are shown in Table IV. There are two important findings: with p_t given, attention module boost neural controller's performance even trained with D_{II}; with s' given to attention module, both ANNet and DANNet reach 100% success rate. Comparing with RF in Table III, the ANNet and DANNet provides on par or even better performance.

TABLE IV: Success rates of the three proposed neural network controllers: NNetV2, single attention network ANNet and double attention network DANNet (see Section III-A). $\vec{s}'$ denotes usage of the additional attention sensors (Table I).

Controller	Training data (summer)			
	p_t	$\vec{s}'$	D_I	D_{II}
NNetV2			56%	0%
ANNet			48%	0%
DANNet			24%	0%
NNetV2	✓		76%	0%
ANNet	✓		100%	60%
DANNet	✓		100%	100%
NNetV2	✓	✓	0%	0%
ANNet	✓	✓	100%	80%
DANNet	✓	✓	100%	100%

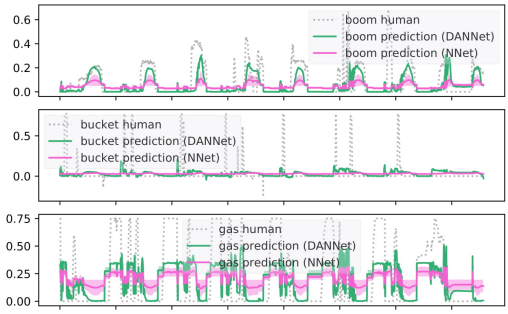


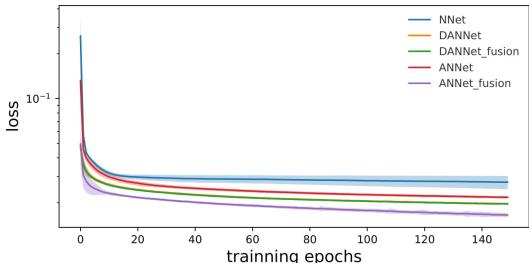
Fig. 5: Comparison of the output control signals for the proposed neural network controllers NNetV2, DANNet and the ground truth (human demonstration).Additional figures are provided within the supplementary video

F. Experiment 4: Simulation studies

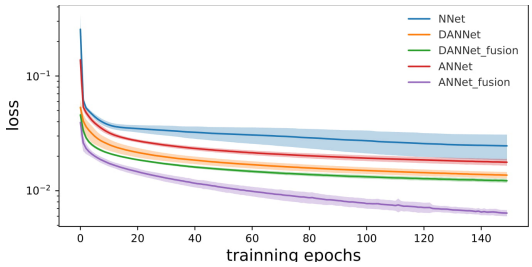
10 recorded human demonstrations in winter conditions allow us to study behavior of the controllers more analytically. Fig. 6 shows the validation loss during training. Two important findings that verify the results with the real pile loader: 1) the MSE loss between the predicted and ground truth control signals obtains smaller test set error with all available training data (Fig. 6(a) and (b)); 2) the attention networks, ANNet and DANNet, always obtain lower MSE than the networks without attention and the attention networks' converge is more stable. These findings can be qualitatively verified in Fig. 5 where the both ANNet and DANNet produce control signals that match the human demonstrations better than NNetV2.

V. CONCLUSION

This work presents new results and findings for learning a pile loading controller from human demonstrations. The previously proposed neural network controller NNet [1] and random forest controller RF [2] fail if the test conditions



(a) Trained with the ideal dataset D_{II} in [2]



(b) Trained with all data D_I

Fig. 6: The mean (solid line) and standard deviation (shading) test data error (validation loss) values for the proposed neural controllers. Values are computed over 100 trials using the same summer training data, random initialization and the same offline winter test data (10 human demonstrations). ANNet_fusion and DANNet_fusion refers to training with s'

are drastically different from the training conditions (see Experiment 1). The failures of the neural controller can be fixed by using a more expressive architecture (NNetV2) and by adopting modern deep learning optimization and non-linearities (Section III). In addition, NNetV2 benefits from more data and from a sensor that makes the control problem more observable (Experiment 2). Finally, the neural network controllers adapting the attention mechanism, ANNet and DANNet, achieve superior results as compared to NNetV2 (Experiment 3). The attention network controllers produce signals that match more accurately to human behavior (Experiment 4). Overall, we are convinced that the proposed attention network controllers ANNet and DANNet are suitable for the task of learning pile loading from demonstrations. Our future work will focus on performing high level tasks that are autonomously learned using reinforcement learning on top of the low level controllers learned by demonstrations.

REFERENCES

[1] E. Halbach, J.-K. Kämäräinen, and R. Ghabcheloo, "Neural network pile loading controller trained by demonstration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[2] W. Yang, N. Strokina, N. Serbenyuk, R. Ghabcheloo, and J.-K. Kämäräinen, "Learning a pile loading controller from demonstrations," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020.

[3] S. Dadhich, U. Bodin, F. Sandin, and U. Andersson, "Machine learning approach to automatic bucket loading," in *24th Mediterranean Conference on Control and Automation (MED)*, 2016.

[4] F. E. Sotiropoulos and H. H. Asada, "Autonomous excavation of rocks using a gaussian process model and unscented kalman filter," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2491–2497, 2020.

[5] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *ICML*, 2014.

[6] S. Dadhich, F. Sandin, U. Bodin, U. Andersson, and T. Martinsson, "Field test of neural-network based automatic bucket-filling algorithm for wheel-loaders," *Automation in Construction*, vol. 97, pp. 1 – 12, 2019.

[7] R. Sutton, D. McAllister, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *NeurIPS*, 2000.

[8] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *the 4th International Conference on Learning Representations, (ICLR)*, 2016.

[9] H. A. H.A. Fernando, J. Marshall and J. Larsson, "Towards controlling bucket fill factor in robotic excavation by learning admittance control setpoints," in *Springer Proceedings in Advanced Robotics 5*, pp. 35–48, 2018.

[10] F. E. Sotiropoulos and H. H. Asada, "A model-free extremum-seeking approach to autonomous excavator control based on output power maximization," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1005–1012, 2019.

[11] P. L. D. Jud, G. Hottiger and M. Hutter, "Planning and control for autonomous excavation," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2151–2158, 2017.

[12] M. N. R. Fukui, T. Niho and M. Uetake, "Imitation-based control of automated ore excavator to utilize human operator knowledge of bedrock condition estimation and excavating motion selection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5910–5916, 2015.

[13] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1955.

[14] V. Mnih, N. Heess, A. Graves, and k. kavukcuoglu, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems 27*, pp. 2204–2212, Curran Associates, Inc., 2014.

[15] S. Jetley, N. A. Lord, N. Lee, and P. Torr, "Learn to pay attention," in *Proceedings of the International Conference on Learning Representations, ICLR*, 2018.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations, ICLR*, 2015.

[17] Y. Rao, J. Lu, and J. Zhou, "Attention-aware deep reinforcement learning for video face recognition," in *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, 2017.

[18] S. G. Venkatesh and B. Amrutur, "One-shot object localization using learnt visual cues via siamese networks," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6700–6705, IEEE, 2019.

[19] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," *arXiv preprint arXiv:1908.03265*, 2019.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.