ExACT: An End-to-End Autonomous Excavator System Using Action Chunking With Transformers

Liangliang Chen, Shiyu Jin, Haoyu Wang, Liangjun Zhang

***Abstract*— Excavators are crucial for diverse tasks such as construction and mining, while autonomous excavator systems enhance safety and efficiency, address labor shortages, and improve human working conditions. Different from the existing modularized approaches, this paper introduces ExACT, an end-to-end autonomous excavator system that processes raw LiDAR, camera data, and joint positions to control excavator valves directly. Utilizing the Action Chunking with Transformers (ACT) architecture, ExACT employs imitation learning to take observations from multi-modal sensors as inputs and generate actionable sequences. In our experiment, we build a simulator based on the captured real-world data to model the relations between excavator valve states and joint velocities. With a few human-operated demonstration data trajectories, ExACT demonstrates the capability of completing different excavation tasks, including reaching, digging and dumping through imitation learning in validations with the simulator. To the best of our knowledge, ExACT represents the first instance towards building an end-to-end autonomous excavator system via imitation learning methods with a minimal set of human demonstrations. The video about this work can be accessed at https://youtu.be/NmzR_Rf-aEk.**

I. Introduction

Excavators are highly versatile heavy equipment, crucial for applications ranging from mining and construction to environmental restoration and emergency rescue, capable of operating in diverse and challenging conditions [1]. Autonomous excavators offer a critical solution to the dangers and inefficiencies of traditional excavation [2]. By operating independently, these machines significantly enhance safety, mitigating the high number of injuries and fatalities due to hazardous excavation environments. They are especially beneficial in remote and harsh locations, where extreme conditions and a scarcity of skilled operators challenge productivity. Autonomous systems also circumvent human limitations, such as fatigue, and address workforce issues like aging and labor shortages [3]. To sum up, autonomous excavators promise to not only improve safety and operational throughput but also provide a sustainable response to labor challenges in the construction and mining industries.

The existing autonomous excavator systems execute modular tasks in sequential order, i.e., perception, prediction, planning, and control [2], [3]. Zhang *et al.* [3] designed an autonomous excavator system for material handling in challenging environments such as mining and construction, utilizing a robust architecture that integrates multimodal perception sensors like LiDAR and cameras with advanced image processing and task planning algorithms. However, the hydraulic excavator is a highly nonlinear system in which the effects of working load, control input delay, and dead zone are hard to capture by the physical models in modular methods. Data-driven methods overcome this issue by directly deriving a model or controller from collected data without physical modeling [4]. Many works have investigated the data-driven modeling or control strategies for excavator systems [5], [6]. The learning-based strategy is an important type of data-driven method that is being actively explored in the field of autonomous excavator systems. Ref. [7] investigated using reinforcement learning to obtain end-effector trajectory tracking controllers for hydraulic excavators. Jin *et al.* [8] developed an offline reinforcement learning excavator controller based on implicit Q-learning [9], which does not require online interactions between the controller and environment.

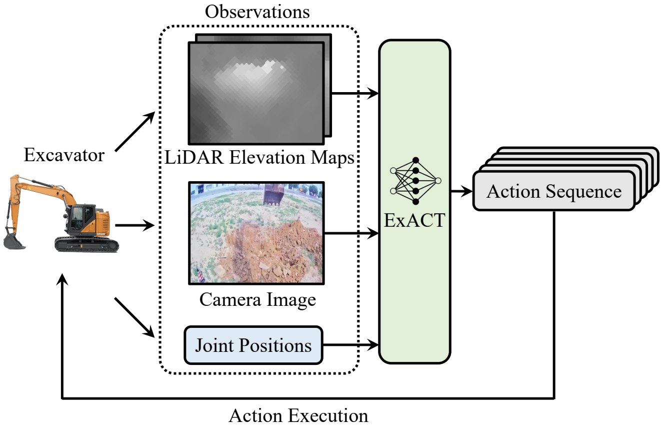


Fig. 1: Framework of ExACT

Recently, efficient imitation learning-based methods have been developed for robot manipulation [10], [11]. Ref. [10] developed the diffusion policy, a robot imitation learning algorithm based on the diffusion model [12]. However, the inference time of the diffusion policy may be too long to satisfy the requirement of high-frequency excavator control. Zhao *et al.* [11] proposed another imitation learning method ACT based on the conditional variational autoencoder [13], [14] and the transformer architecture [15].

This paper proposes ExACT, an end-to-end autonomous excavator system, which takes the raw LiDAR and camera inputs and directly output valve command to control the ex-

Robotics and Autonomous Driving Lab, Baidu Research, USA. liangjunzhang@baidu.com. This work was done when Liangliang Chen was an intern at Baidu Research.

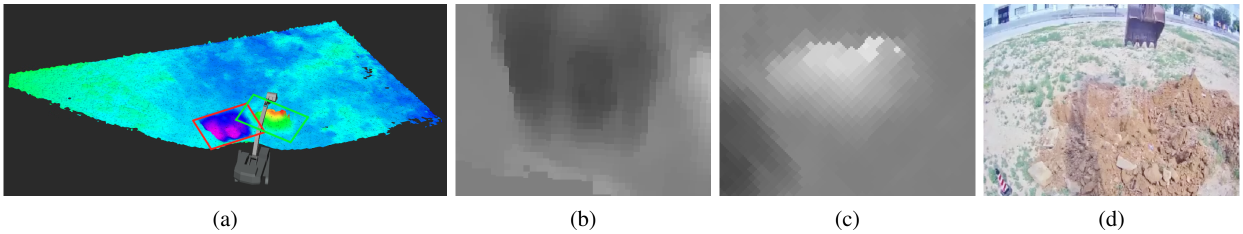


Fig. 2: Examples of the front camera image and LiDAR elevation maps. (a) Raw LiDAR elevation map (visualized in RViz) from which the digging (red box) and dumping (green box) LiDAR elevation maps are cropped; (b) Preprocessed LiDAR elevation map of the digging zone; (c) Preprocessed LiDAR elevation map of the dumping zone; (d) front camera image.

cavator. Our system leverages ACT architecture for imitation learning. Utilizing observations from multi-modal sensors such as LiDAR, cameras, and inclination sensors, ExACT generates a sequence of actions that can be executed sequentially. With only a limited number of human demonstration data collected from a real excavator, we show that the excavator can complete a set of tasks through imitation learning with only a few comprehensive demonstration trajectories. To the best of our knowledge, this is the first instance towards building an end-to-end autonomous excavator system via imitation learning methods with a minimal set of human demonstrations.

II. ExACT: An End-to-End Autonomous Excavator System

The proposed autonomous excavator system, ExACT, leverages ACT to train an excavator controller by imitation learning. ACT is an imitation learning algorithm that has demonstrated excellent performance on end-to-end bimanual robotic manipulations [11], [16]. Different from the traditional imitation learning methods, such as behavioral cloning [17], ACT addresses the compounding errors by using action chunking, i.e., predicting a sequence of actions, and temporal ensembling to generate smooth actions. In addition, ACT leverages a conditional variational autoencoder [14] at a high level to generate action sequences. Employing a generative model strengthens the controller's robustness against the noise in human demonstration data.

The method framework is presented in Fig. 1. The controller inputs consist of a camera image and the excavator joint positions. Unlike the original ACT algorithm [11], ExACT considers the LiDAR elevation maps of the digging and dumping areas as additional inputs if the excavator task involves digging and dumping. The controller outputs a sequence of actions that can be implemented on the excavator to fulfill the task. We use temporal ensembling to make the actions smooth. With this strategy, we make an action sequence prediction at each timestep and ensemble the relevant actions by the exponential weighting scheme. The obtained ensembled actions are implemented in the excavator. The experiment details can be found in Section III.

For the digging and dumping tasks, the 3D geometry information of the digging and dumping zones is important for the excavator controller to decide where to dig and dump. The LiDAR equipment detects the elevations of the workspace, from which we can extract the elevation maps of the digging and dumping zones. We cropped two local elevation maps from the raw LiDAR elevation map that provides ExACT with accurate 3D geometry information of the digging and dumping zones.

III. Experiments

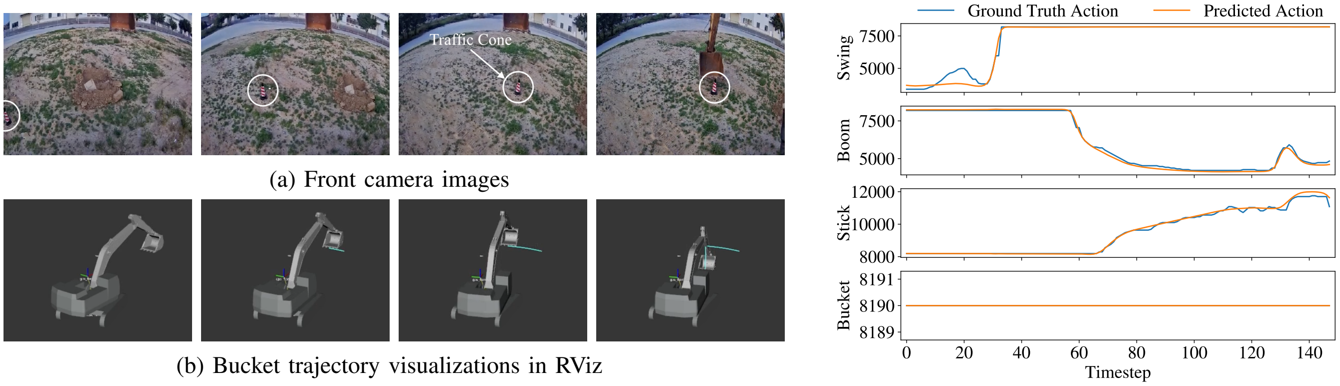
In the experiments, the robotic platform employs a 21.5-ton hydraulic excavator that has been upgraded with a drive-by-wire system. This allows the excavator to be operated either through a handheld remote control (RC) or controlled via computer using a CAN bus interface. The excavator is further equipped with a variety of sensors, including inclinometers for measuring the joint pose of the excavator's boom, stick, and bucket, an encoder for measuring the swing angle of the cabin, a 3D LiDAR sensor, multiple RGB cameras, an IMU, and a real-time kinematic (RTK) GPS. The excavator is installed with an industry-standard computer with a GPU.

A. *Excavator Task Descriptions*

The following excavator tasks are considered in our experiments.

1) `reach`: Use the excavator bucket to reach a fixed target position, indicated by a traffic cone, from different locations. The controlled variables of this task are the valve states, which means that this is an end-to-end control task.
2) `dig_dump`: Dig soil in a certain area and dump it in another area. We consider the valve states as the controlled variables for this task.
3) `dig_dump_return`: Dig soil in a certain area, dump it to another area, swing the cabin, and return the bucket to a position above the digging area for the next circle of digging. The controlled variables for this task are the joint positions.

The input variables of all the tasks above include: i) the 4-dimensional joint positions (i.e., swing, boom, stick, and bucket); ii) a front camera image with the size $480 \times 640 \times 3$; iii) the LiDAR elevation maps of the digging and dumping zones with the sizes $480 \times 640 \times 3$, respectively, if the task involves digging and dumping. The examples of the LiDAR elevation maps and the front camera image are shown in Fig. 2.



(a) Front camera images

(b) Bucket trajectory visualizations in RViz

(c) Ground truth and predicted actions

Fig. 3: Test performance of the task `reach` (valve state control)

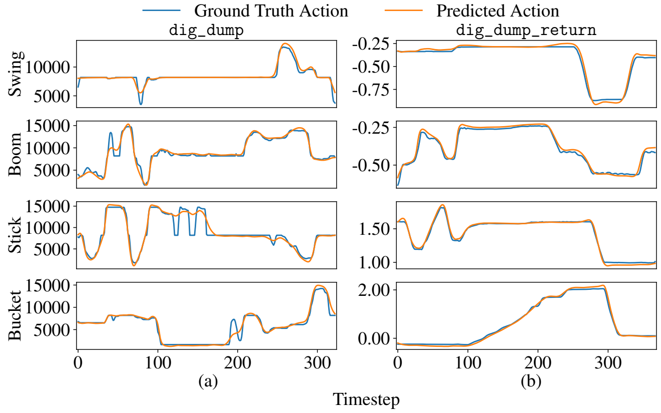


Fig. 4: Ground truth and predicted actions of the tasks (a) `dig_dump` and (b) `dig_dump_return`

B. Data Collections

We control the valve states of the excavator remotely, which determines the joint velocities, although with the time delay and dead zone effects. The data frequency is 10 Hz. TABLE I shows the statistics of the collected data, including the number of episodes and average episode length in terms of timesteps. In all tasks, we randomly select one episode as a test episode and use all the other episodes as training data.

C. Valve States to Joint Velocities Modelling

In our excavator experiment, four valve states are controlled directly to adjust the 4 joint velocities, i.e., swing, boom, stick, and bucket. However, it is unsafe to directly test ExACT on the real excavator without any offline pretest. In our experiments, this offline pretest is achieved by converting the valve states to the corresponding joint velocities via an approximate linear model. Based on the 2661 data of valve states-joint velocities pairs, the linear models are derived as

$$q^{\mathrm{d}}_{\mathrm{swing}} = -2.8227 \times 10^{-6} \cdot V_{\mathrm{swing}} + 2.3118 \times 10^{-2}, \qquad (1a)$$

$$q^{\mathrm{d}}_{\mathrm{boom}} = 1.3736 \times 10^{-6} \cdot V_{\mathrm{boom}} - 1.1250 \times 10^{-2}, \qquad (1b)$$

$$q^{\mathrm{d}}_{\mathrm{stick}} = -2.4656 \times 10^{-6} \cdot V_{\mathrm{stick}} + 2.0193 \times 10^{-2}, \qquad (1c)$$

$$q^{\mathrm{d}}_{\mathrm{bucket}} = 5.8151 \times 10^{-6} \cdot V_{\mathrm{bucket}} - 4.7625 \times 10^{-2}, \qquad (1d)$$

TABLE I: Data statistics of different tasks

Task	Number of episodes	Average episode length
`reach`	8	180.5 timesteps
`dig_dump`	12	361.0 timesteps
`dig_dump_return`	12	397.8 timesteps

TABLE II: Training hyperparameters

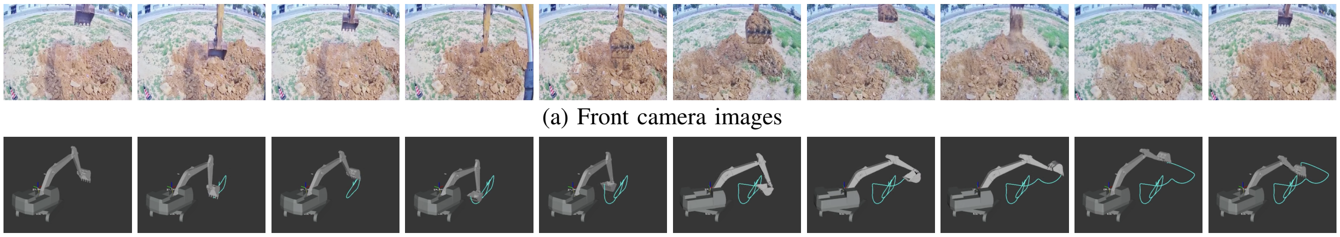
Hyperparameter	Value
KL divergence weight	10
Chunk size	30
Number of steps	30000
Learning rate	1.0×10^{-5}

where q^{d}_j and V_j, with $j \in \{\mathrm{swing}, \mathrm{boom}, \mathrm{stick}, \mathrm{bucket}\}$, denote the joint velocities and valve states of the corresponding joints, respectively. Note that the default value of all valve states is 8190, under which the corresponding joint velocity is 0.

D. Results

With the task and data descriptions in Sections III-A and III-B, we train ExACT with the hyperparameters in TABLE II. In the test phase, we initialize the joint positions of the first step as the true values in the test episode. The remaining joint positions are calculated based on the corresponding previous joint states and actions. This strategy effectively avoids the phenomenon that the joint positions output by ExACT track the previous real joint positions. For the front camera image and LiDAR elevation maps, we use the real collected data from the test episode. The temporal ensembling method [11] is utilized during the test in all tasks to smooth the trajectory.

1) Results of the `reach` Task: After training ExACT with 8 episodes of demonstration data, we obtain the test results shown in Fig. 3. In the data collection process, we first align the bucket vertically with the traffic cone and then move down to the target position. Figs. 3(a)-(b) demonstrate that ExACT learns the correct reaching behavior, and the bucket first moves horizontally and then vertically. The success of the `reach` task also indicates the linear models of valve states and joint velocities, as shown in Section III-C, are



(a) Front camera images

(b) Bucket trajectory visualizations in RViz

Fig. 5: Front camera images and bucket trajectory during the testing of the task `dig_dump_return` (joint position control)

effective when the excavator task is straightforward. Fig. 3(c) compares the ground truth and predicted valve states. We note that the predicted value states align well with the ground truths, which further validates the excellent performance of ExACT on the excavator `reach` task.

2) Results of the `dig_dump` Task: The ExACT test results of the `dig_dump` task after training with 11 episodes are shown in Fig. 4(a), from which we can find that ExACT struggles to learn the high-frequency components of the valve states. Since the valve states directly impact the joint velocities, the failure to learn the high-frequency components of the valve states will lead to inflexible joint movements. This may cause the bucket edge to move longer than it should during the digging and dumping process in practice. We leave the development of an ExACT controller that can learn the high-frequency components for future work. Moreover, more demonstration data might be collected to improve the ExACT performance on the `dig_dump` task with a valve state controller.

3) Results of the `dig_dump_return` Task: Due to the imperfect performace of valve state control in the `dig_dump` task, we use the joint position control in the `dig_dump_return` task, which include one additional phase of `return`. The task visualizations and action predictions during the test are shown in Figs. 4(b) and 5, respectively. The results demonstrate that the excavator can complete the whole task perfectly, with the joint positions being predicted with high precision.

IV. Conclusions

This paper presents ExACT, an end-to-end autonomous system that leverages raw data from LiDAR and cameras and joint positions to control the movements of an excavator. Employing the ACT architecture, this system integrates imitation learning with multi-modal sensor data to produce sequences of actionable commands. Our experimental setup included a simulator developed using linear equations to represent the dynamics between the excavator valve states and its joint velocities. With a limited number of human demonstration trajectories, ExACT successfully executed a variety of excavation tasks in this simulated environment. This achievement marks a significant milestone in the excavator learning field as, to the best of our knowledge, it is the first instance towards building an imitation learning-based end-to-end excavator controller with minimal human demonstrations. In the future, we plan to test the performance of ExACT on real excavators. Our work paves the way for further research and development in autonomous excavation technologies.

References

[1] R. L. Johns, M. Wermelinger, R. Mascaro, D. Jud, I. Hurkxkens, L. Vasey, M. Chli, F. Gramazio, M. Kohler, and M. Hutter, "A framework for robotic excavation and dry stone construction using on-site materials," *Science Robotics*, vol. 8, no. 84, p. eabp9758, 2023.

[2] O. M. U. Eraliev, K.-H. Lee, D.-Y. Shin, and C.-H. Lee, "Sensing, perception, decision, planning and action of autonomous excavators," *Automation in Construction*, vol. 141, p. 104428, 2022.

[3] L. Zhang, J. Zhao, P. Long, L. Wang, L. Qian, F. Lu, X. Song, and D. Manocha, "An autonomous excavator system for material loading tasks," *Science Robotics*, vol. 6, no. 55, p. eabc3164, 2021.

[4] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.

[5] R. J. Sandzimier and H. H. Asada, "A data-driven approach to prediction and optimal bucket-filling control for autonomous excavators," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2682–2689, 2020.

[6] M. Lee, H. Choi, C. Kim, J. Moon, D. Kim, and D. Lee, "Precision motion control of robotized industrial hydraulic excavators via data-driven model inversion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1912–1919, 2022.

[7] P. Egli and M. Hutter, "A general approach for the automation of hydraulic excavator arms using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5679–5686, 2022.

[8] S. Jin, Z. Ye, and L. Zhang, "Learning excavation of rigid objects with offline reinforcement learning," *arXiv preprint arXiv:2303.16427*, 2023.

[9] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," in *International Conference on Learning Representations*, 2022.

[10] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *arXiv preprint arXiv:2303.04137*, 2023.

[11] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Robotics: Science and Systems 2023*, Daegu, Republic of Korea, 2023, pp. 1–19.

[12] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[13] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[14] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[16] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024.

[17] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Advances in Neural Information Processing Systems*, vol. 1, 1988.