Robot Learning for Manipulation of Granular Materials Using Vision and Sound

Samuel Clarke

June 2019
CMU-RI-TR-19-68

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania

Thesis Committee:
Christopher G. Atkeson, *Co-Chair*
Oliver Kroemer, *Co-Chair*
David Held
Rogério Bonatti

*Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Robotics.*

Abstract

Granular materials are ubiquitous in household and industrial manipulation tasks, but their dynamics are difficult to model analytically or through simulation. During manipulation, they provide rich multimodal sensory feedback. We present a robotic system we constructed for investigating manipulation of granular materials. We present two data-driven, learning-based frameworks to control scooping and pouring granular materials.

In our first set of experiments, we focus on the task of scooping granular materials and propose to learn a model of relevant granular material dynamics through a data-driven approach based on neural networks. We evaluate our approach on a dataset of 7,380 samples of scooping actions with two different granular materials. Our results indicate that our model is effective for predicting both mass scooped and change in the material's height map, with a respective mean RMSE of 5.8 g and 0.38 cm on each task. We also demonstrate how our model may be used for control of scooping a desired mass.

In our second set of experiments, we demonstrate a novel framework for using audio feedback during manipulation of granular materials. Granular materials produce audio-frequency mechanical vibrations in air and structures when manipulated. These vibrations correlate with both the nature of the events and the intrinsic properties of the materials producing them. We therefore propose learning to use audio-frequency vibrations from contact events to estimate the flow and amount of granular materials during scooping and pouring tasks. We evaluated multiple deep and shallow learning frameworks on a dataset of 13,750 shaking and pouring samples across five different granular materials. Our results indicate that audio is an informative sensor modality for accurately estimating flow and amounts, with a mean RMSE of 2.8 g across the five materials for pouring. We also demonstrate how the learned networks can be used to pour a desired amount of material.

Acknowledgments

My advisors, Chris Atkeson and Oliver Kroemer, went above and beyond in their support – financially, technically, and morally. None of this would have been possible if they had not entrusted me with an undue amount of freedom and responsibility, all the while backing me up with great mentorship in all realms through technical contributions, personal character development, and life direction.

My other committee members, David Held and Rogério Bonatti, not only reviewed this thesis, but also similarly supported me the past two years whenever I needed advice.

Much of this work was completed in collaboration with Travers Rhodes, who has been a role model to me of positivity and humility during my time here.

My lab mates, especially Maximilian Sieb and Austin Wang, were always there to help me when I was stuck and spurred me on to keep up with their pace.

I could not have asked Tim Angert and Chuck Whittaker to be more helpful than they were whenever I had mechanical and machining needs. All the hours they put into training me in how to use the tools effectively and giving me manufacturing advice were invaluable.

My family has been laying foundations for me since before I was born. They have gotten me through the times where I thought the light at the end of the tunnel might just be a mirage, then they have been there to rejoice with me when I made it to the other end.

If I listed off all the friends that have supported me through this, it would go on to the next page, but thank you for the long phone calls, the dinners, and the weekend diversions.

As a deep believer in God, I don't believe any of these opportunities and the relationships that begat them would have been possible if I had not been put here for a reason. I see the research process as the act of uncovering underlying structure that has been artfully put in place since the beginning by a creator. Throughout this process, the most meaningful moments have been when I felt I was seeing God move to provide me with ideas, reality

checks, opportunities, and all these amazing relationships.

This material is based upon work supported in part by the National Science Foundation under grant IIS-1717066.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Granular materials permeate our everyday lives, from the food we eat to the ground on which our homes are built. Whether we are serving ourselves cereal for our morning breakfast or repotting a house plant, manipulating granular materials in our environment is frequently required in daily human tasks. Granular materials also play important roles in industrial processes, such as ground excavation for mining, or mixing for glass and plastic production [7]. In order for robots to assist in many household and industrial tasks, they must be able to effectively manipulate granular materials.

The domain of granular materials presents challenging robotic manipulation problems since manipulation typically relies on a model of the environment to predict the results of different actions. The dynamics of granular materials depend on the surface interactions among many constituent particles with potentially heterogeneous individual shapes. These complex dynamics are often very difficult to analytically model and can be expensive to simulate [1].

In order to investigate the challenges of robotic manipulation of granular materials, we designed a robotic setup, shown in Figure 1.1. The setup was designed for flexibility among multiple experiments, two of which will be detailed in this thesis. The setup includes a

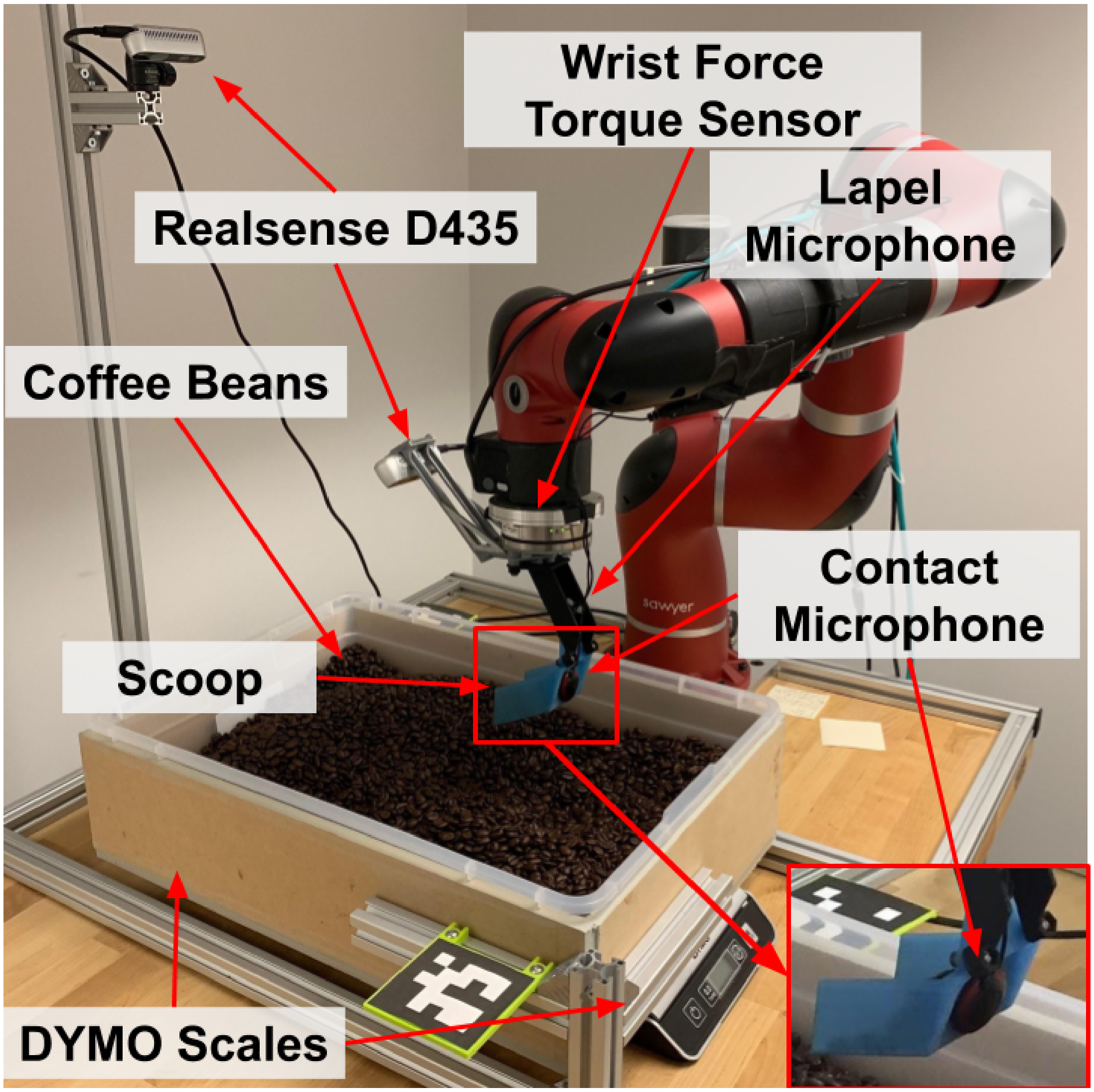


Figure 1.1: Robotic setup for investigating manipulation of granular materials with multimodal sensory feedback. The tub in front of the Sawyer 7-DOF robot arm can be exchanged for tubs with different granular materials.

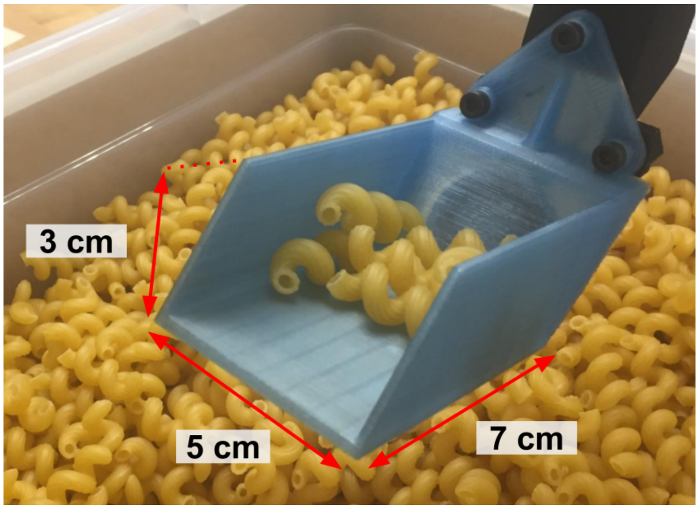


Figure 1.2: The scoop end effector used in our experiments.

Sawyer 7-DOF arm mounted to a table. A tub of a granular material rests in front of the robot on top of two Dymo USB postage scales for detecting changes in the mass of material in the tub. On the end of the robot is a custom 3D printed scoop end-effector, with a close-up image shown in Figure 1.2.

The setup is outfitted with numerous sensors for sensing multiple modalities of signals. A Realsense D435 camera measuring RGB and depth images captures the surface of the granular material, and an additional D435 camera is mounted to the end-effector, facing down into the scoop to capture the RGB and depth of the contents of the scoop. Between the wrist and the scoop is an ATI Axia80 wrist force-torque sensor for sensing the forces and moments on the scoop during manipulation. The handle of the scoop has a lapel microphone with a cardioid profile pointing toward the tip of the scoop's basin for recording airborne sound. Finally, a contact microphone is glued to the back of the scoop's basin for measuring structural vibrations from the body of the scoop during manipulation.

For testing the generality of the frameworks we test in our experiments, we selected different granular materials to manipulate, shown in Figure 1.3. Each material is relevant to a different application of manipulating granular materials, with applications of robotic

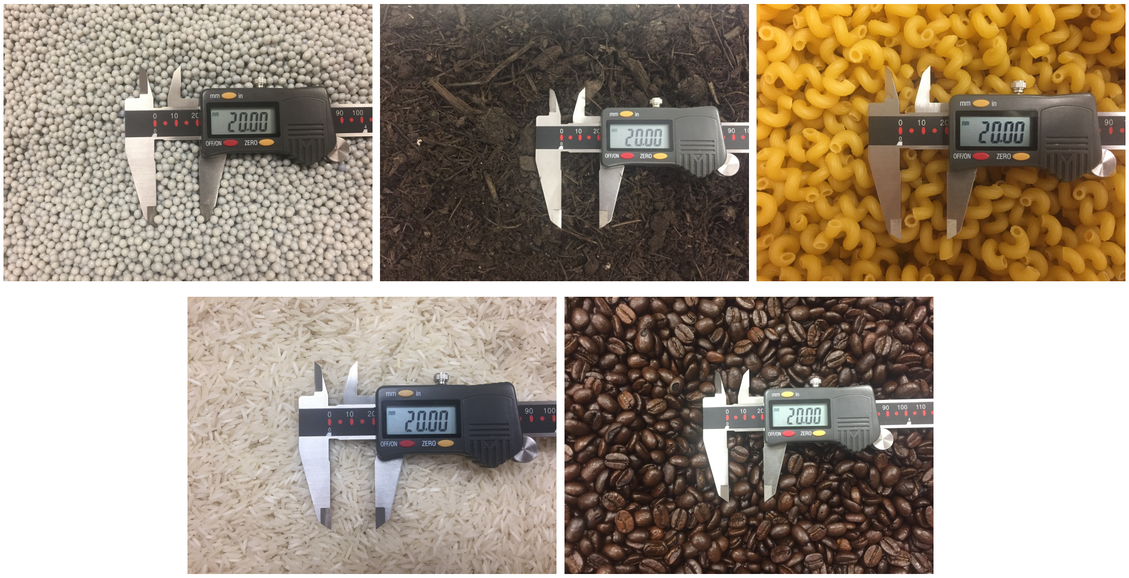


Figure 1.3: The granular materials used in our experiments, with a 2 cm-wide calipers for scale. **(Top Left)** The plastic injection molding pellets ("Pellets"). **(Top Center)** The peat moss top soil ("Soil"). **(Top Right)** The cellentani pasta ("Pasta"). **(Bottom Left)** The long grain Basmati rice ("Rice"). **(Bottom Right)** The coffee beans ("Coffee").

manipulation in the household, the field, and the factory all being represented. Each of the materials has very different properties in terms of density, granule stiffness, acoustic conductivity, *etc.*, which have different effects on the approaches we must use for manipulation.

Chapter 2

Learning to Predict the Effects of Scooping Granular Materials from Local Height Maps

2.1 Introduction

In this chapter we investigate learning models to approximate the dynamics of different granular materials during robotic manipulation tasks. We focus on the task of a robot scooping a desired mass from a container of material, given an observation of the surface of the material. More specifically, we focus on predicting the parameters that define the desired trajectory for the scooping skill. We also explore predicting the change in shape of the material given different scooping actions.

We evaluate our method using data from scooping with a 7-DOF Sawyer robot arm from Rethink Robotics with a plastic scoop as an end effector (Figure 1.1, 1.2). We evaluate our approach with two different granular materials: plastic injection molding beads and dry cellentani pasta. These two materials represent different potential robotics applications, but perhaps more importantly they represent different material dynamics due to the structures of their constituent particles. Whereas the injection molding pellets are small, smooth, elliptical, and convex, cellentani pasta pieces are large, helical, and potentially interlocking.

2.2 Problem Definition

Each scooping action by the robot is parameterized by 6 distinct parameters, $(x, y, z_\text{i}, z_\text{f}, L, \theta)$. The x and y coordinates define where the tip of the scoop will begin its plunge in the container; z_i and z_f are the respective initial and final height of the scoop through its sweep; L is the length of the scooping sweeping motion; and θ is the fixed angle the scoop maintains through the motion. These parameters are each visualized in Figure 2.1. Before performing a scoop, the robot observes a discretized height map H of the surface of the material. From these scooping parameters and the observed height map, the robot should be able to estimate the mass m_est of granular material scooped as well as predict the resulting surface height map H'_est. We reduce the dimensionality of the input to our model by centering H on the x and y values. The height map is thus defined relative to the starting location of the scoop. Our model may be described as:

$$\{m_\text{est}, H'_\text{est}\} = f(H, z_\text{i}, z_\text{f}, L, \theta) \tag{2.1}$$

We optimize our model with respect to a mean squared error loss on its predictions. We evaluate our framework against a geometric baseline which assumes all material within the trajectory of the scoop will be removed without otherwise affecting its surroundings.

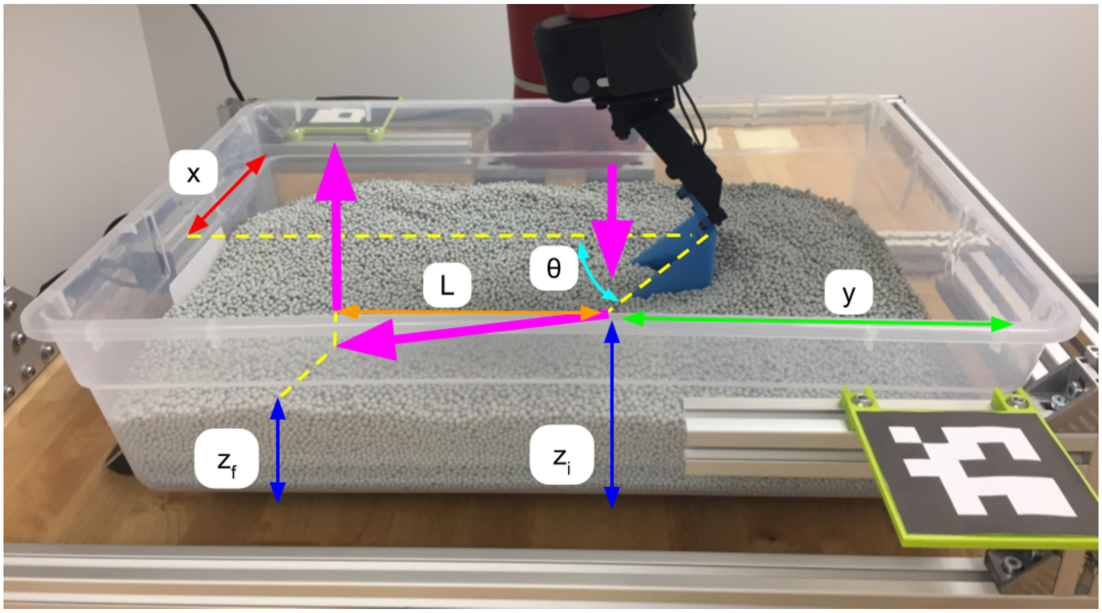


Figure 2.1: The parameters of the scooping action. The scooping motion can be described as follows: the robot hovers the scoop above the tub at the (x, y) coordinates, sets the pitch angle of the scoop to θ, then plunges the scoop straight down to height z_{i}. It then follows a linearly interpolated path to point $(x, y + L, z_{\mathrm{f}})$ while retaining the same angle θ. Upon reaching this final point, the robot tilts the scoop up about its back edge to a constant angle of $\pi/12$ from horizontal in order to retain the scooped material, and then lifts up out of the tub. In the example shown in the figure, the trajectory of the tip of the scoop would follow the magenta arrows.

2.3 Related Work

Engineering models of granular materials have been developed for recovering mechanical phenomena and properties from simulation. The authors of [41] and [22] found that modelling granular materials using discrete element methods (DEM) as heterogeneous 2D or 3D ellipsoids rather than as as circles or spheres made for higher simulation quality. The authors of [14] use a hierarchical framework coupling a finite element method and DEM techniques to model the behavior of granular materials at a high and a low level, respectively. They demonstrate their technique to be effective in modeling stresses and strains and localizing them within a large body of sand.

Simulation of granular materials has been refined in the computer graphics community [39]. For example, the authors of [47] simulated sand behavior by beginning with a water simulator and accounting for dynamics more specific to granular materials, such as interparticle friction to achieve a "visually acceptable result." The authors of [25] found that dispensing with many fluid-based assumptions in modelling granular materials and solving for the internal frictional stresses within the material allowed graphical simulations to reproduce even more realistic behavior.

With respect to robots and granular media, there is a sizable body of work on mobile robots interacting with granular media while moving or locomoting [20, 21]. Some work has approached building models and simulations specifically for modeling these types of interactions [23]. There is also work in modeling and controlling scooping motions for large construction equipment [30]. The authors of [15] compare using genetic algorithms and self-learning simulations, wherein a learned neural network-based model of soil properties is combined with finite element analysis, to predict soil deformations during excavation. In [35], the author compares several different models for predicting the resistive forces experienced by a robotic excavator digging through soil. He develops an analytical model of a flat blade moving through soil and compares several learning-based models, including neural networks, in their ability to tune analytical models to more closely match observed data. Whereas the excavation application usually concerns granular materials with particles much smaller than the manipulator, we will model the interactions between a manipulator and granular particles within an order of magnitude in size.

Data-driven or learned models for robotic manipulation have become increasingly popular. Paolini et al. combined kernel density estimators and Gaussian processes to model the probability of grasp success [26]. Deep learning based models, where large neural networks are trained on large datasets, have also proven to be quite effective in modelling complex dynamics in many real-world robotics domains. [19, 27] and [9] have successfully used convolutional neural network (CNN) architectures for modelling dynamics in robotics applications from grasping to drone flying. CNN architectures consist of sequentially applying convolutional kernels to inputs, with weights learned through gradient descent. Their convolutional nature lends itself well to extracting features from inputs where spatial features are translationally invariant.

A similar work to our own is that of Schenck et al., in which a robot learned to scoop and dump pinto beans from one container to another to achieve a desired shape [33]. The authors trained a predictive model based on a CNN and used the Cross Entropy Method (CEM) [3] to use their model to sample actions for their policy. An approach similar to Schenck et al. will be used in this paper, in that we will use a height map as input to a convolutional neural network. However, while our own approach uses scoop action parameters directly as inputs to our model, Schenck et al. added an “Action Map” channel for the trajectory of the scoop in image space as input to their fully convolutional model. The height maps we use as input to our network are centered on the location of the beginning of the scooping motion in order to reduce the dimensionality of our inputs.

2.4 Approach

In this section, we present a data-driven approach to construct a model for open-loop control of scooping. We describe the underlying structure of our model as well as how we collect training data.

2.4.1 Granular Material Egocentric Scooping Dataset

We collected training examples for our model using the setup shown in Figure 1.1. The 3D-printed scoop end effector is composed of a nylon handle, which is semiflexible to protect the scoop from breaking, and a rigid polylactic acid (PLA) rectangular basin of 7 cm length, 5 cm width, and 3 cm depth (Figure 1.2). For each scooping example, we sampled action parameters uniformly over the following intervals: $x \in [-11, 11]$ cm from the center of the tub, $y \in [19, 20]$ cm from the right wall of the tub, $L \in [0, 14]$ cm, $z_{\mathrm{i}}, z_{\mathrm{f}} \in [0, 9.5]$ cm from the bottom of the tub, and $\theta \in [\pi/10, \pi/4]$. Each scooping action was performed in the left direction, with the opening of the scoop parallel to the left wall, as shown in Figure 2.1. Before performing each scoop, we collected a 1280×720 RGBD image of the container from the Intel RealSense camera, then took a reading from the scales under the tub of the initial mass of the tub. Note that each scale has a measurement resolution of 2 g.

The scooping action is performed with the robot in Cartesian impedance control mode with maximum stiffness in the x and y direction, and a stiffness of 400 N/m in the z direction to protect the scoop from damage in case of jamming. Thus, the scoop's adherence to its desired trajectory may be affected by the resistive forces of the material while scooping. After performing the scooping action specified by the randomly selected parameters, the scoop was tilted about the back edge of its basin to a pitch of $\pi/12$ from horizontal to retain the material it had scooped. The scoop was lifted straight up at this angle. We then measured the final mass of the tub and subtracted the initial mass to determine the mass of

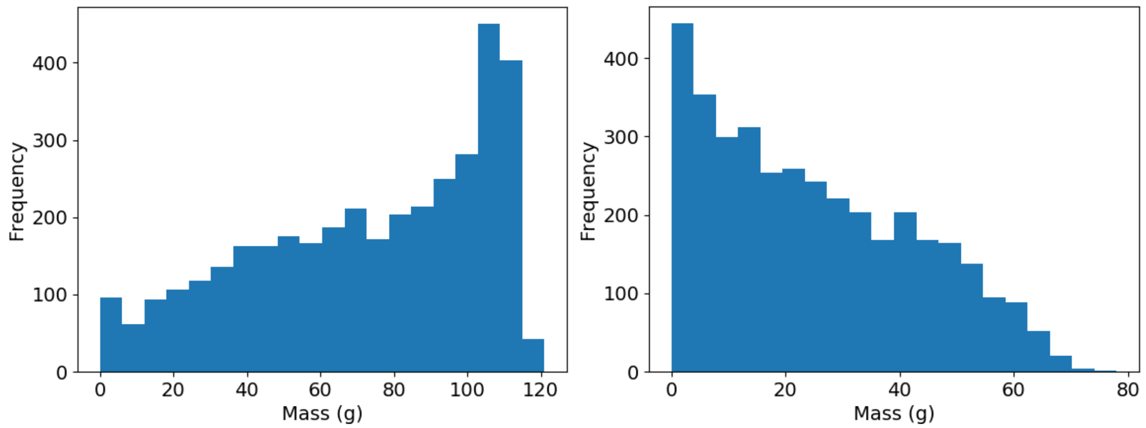


Figure 2.2: The histograms of scooped masses. **(Left)** For the pellets. **(Right)** For the pasta. Differences in the skews of the distributions are likely due to the scoop being more prone to jamming when plunging into and sweeping through the pasta than the pellets.

the granular material scooped by that action. Before dumping the material back into the container, we collected an additional RGBD image of the container to capture the resulting surface of the material after the scooping action.

We collected 3,690 examples from each material. Our setup was able to collect about 120 samples per hour, so this dataset represents approximately 60 robot-hours of data. A histogram of masses scooped for the dataset of each material is shown in Figure 2.2.

2.4.2 Dataset Preprocessing

For each training example, we transformed the points from the camera's RGBD images from the camera frame to the world frame. We then centered the x and y coordinates of the point cloud on the x and y value from the scoop parameters, defining the coordinates relative to the beginning of the scooping trajectory. Finally, we interpolated the depth bilinearly from this point cloud over a discretized 0.5 cm grid from a rectangle spanning 4 cm behind to 20 cm ahead of the beginning of the scoop action on the y axis, as well as 3.5 cm in both directions of the x axis from the center of the scoop. Thus, a 14×48 height matrix was constructed from a 7×24cm section of the granular material. An example of

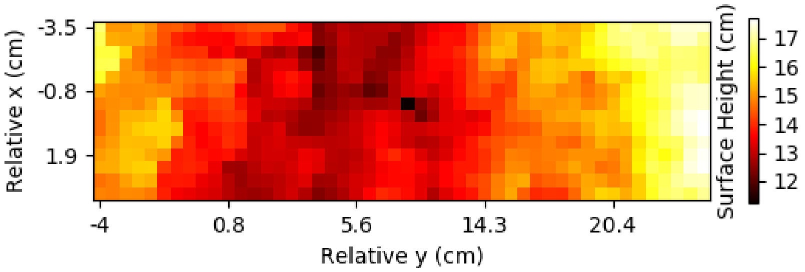


Figure 2.3: An example discretized height matrix H from a pasta surface, constructed by preprocessing. x and y are shown relative to the beginning of the scoop action, with the scoop sweeping in the positive direction of this relative y.

such a height matrix is shown in Figure 2.3.

Centering on the scooping action's starting point reduces the dimensionality in two ways: by obviating the need for the model to use the x and y starting positions of the action and by reducing the size of the height map matrix without sacrificing its resolution. This simplification is based on the assumption that the dynamics of scooping of granular materials are strongly dominated by the structure of the material local to the scooping action.

The only preprocessing performed on the actions was to compute the $\sin(\theta)$ and $\cos(\theta)$ of the scoop angle θ. Thus, the inputs to our model were a 14×48 initial height map H, the initial scoop height z_{i}, the final scoop height z_{f}, the scoop length L, and the angle parameters $\sin(\theta)$, and $\cos(\theta)$. All dimensioned units were in meters. The label of each training example for the mass estimation task was the mass m of material scooped, in grams, and the label for the task of predicting the resulting surface was H', the resulting height matrix.

2.4.3 Model Architecture

The locally-correlated and hierarchical spatial structure of the height maps of granular materials plays to the strengths of a CNN-based approach. Our architecture thus combines

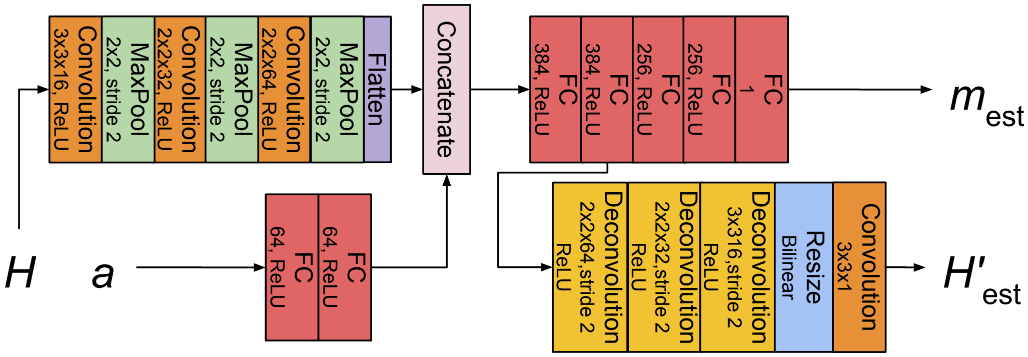


Figure 2.4: The full architecture of our proposed model. In the figure, a represents the vector of preprocessed scooping action parameters.

a CNN structure for automatically extracting features from the height map with a densely connected neural network structure for extracting features from the action parameters. A schematic of the architecture is provided in Figure 2.4.

The height map was first passed through three convolutional layers with $3 \times 3 \times 16$, $2 \times 2 \times 32$, and $2 \times 2 \times 64$ convolutional kernels, respectively. Each convolutional layer was followed by a Rectified Linear Unit (ReLU) activation layer to add nonlinearity, as well as a 2×2 Max Pool layer with stride 2 ($i.e$, the output is reduced to the maximum of every 2×2 contiguous region within each channel). In parallel, the preprocessed action parameters were passed through two densely connected layers of 64 units each and followed by ReLU activations.

The outputs of the convolutional network for the height map and the dense network for the actions were then concatenated together and passed through two densely connected layers, each with 384 units followed by a ReLU activation. The output of the network up to this point was used for both the task of estimating the mass scooped and estimating the resulting height map H'.

For estimating the mass, this intermediate output was subsequently passed through two densely connected layers of size 256 units with ReLU activations. A final simple layer computed a linear function of the output of the penultimate layer to produce the final output of the model for the task of estimating the mass m.

For estimating the resulting height map H', the intermediate output was passed through three deconvolutional layers which each upsample their input with a transpose convolution using a kernel with learned weights. These layers used $2\times2\times64$, $2\times2\times32$, and $3\times3\times16$ kernels, respectively, each with a ReLU activation and a stride of 2 to dilate the intermediate output from $1 \times 6 \times 64$ to $8 \times 48 \times 16$. This was then resized with bilinear interpolation to a size of $14 \times 48 \times 16$, then finally convolved with a $3 \times 3 \times 1$ kernel to condense the output to a single channel and produce an estimate of H'.

Networks were trained separately for the mass estimation and for the height prediction tasks, with gradient descent performed with respect to a mean squared error loss on the corresponding label. During training, the outputs of the densely connected layers were dropped out with probability 0.5 (*i.e.* half of the values were set to 0, and the remaining values were scaled by a factor of 2), to mitigate the risk of overfitting [38].

2.5 Evaluation

We present the results of two experiments for evaluating our method. In the first experiment, we demonstrate the estimation potential and data efficiency of our framework, and in the second experiment, we demonstrate how our framework may be used to generate trajectories for scooping a desired mass.

2.5.1 Prediction Baseline

To evaluate the proposed approach, we use a baseline similar to the heuristic used in [33]. The baseline assumes that all material within the width of the scoop's basin and above the leading edge of the basin within the scooping action's trajectory will be scooped, removing it from the surface of the height map without affecting the untouched regions at all.

For predicting mass, we numerically approximate the volume integral of the scooped mass by multiplying the area of each grid square by the sum of the positive differences in height between the material in each cell and the leading edge of the scoop at each “affected” point in the grid. We then multiply this volume estimate by the measured packing density of each material, 0.94 g and 0.38 g/cm^3 for the pellets and pasta, respectively. For the final baseline estimate, we take the minimum between this value and the maximum payload of the scoop for that granular material.

For predicting the next height matrix H', we assume the final height of any affected point in the matrix will be the expected height of the leading edge of the scoop at that point in its trajectory, and all unaffected points will retain their same heights.

2.5.2 Prediction Accuracy

For evaluating our model’s accuracy, for each material, we first separated a test set of one sixth of the data. We then performed 5-fold cross validation on the remainder of the dataset. To test our model’s sample efficiency, we varied the proportion of the training examples we used, then trained our model through 1000 epochs until the cross validation error on the held out fold reached a minimum. We then computed the error on the test set, using this trained model. The results of this experiment for mass estimation are shown in Figure 2.5, and those for resulting height map estimation are shown in Figure 2.6.

For qualitative inspection, examples of the height map prediction H'_{est} of our network are shown on the left and right of Figure 2.7 for the pellets and pasta, respectively. The results presented are from models that have been trained on 80% of each material dataset over 500 epochs until a minimum error had been achieved on 10% of each dataset held out for cross validation. Results shown are from the remaining 10% of each dataset held out for testing.

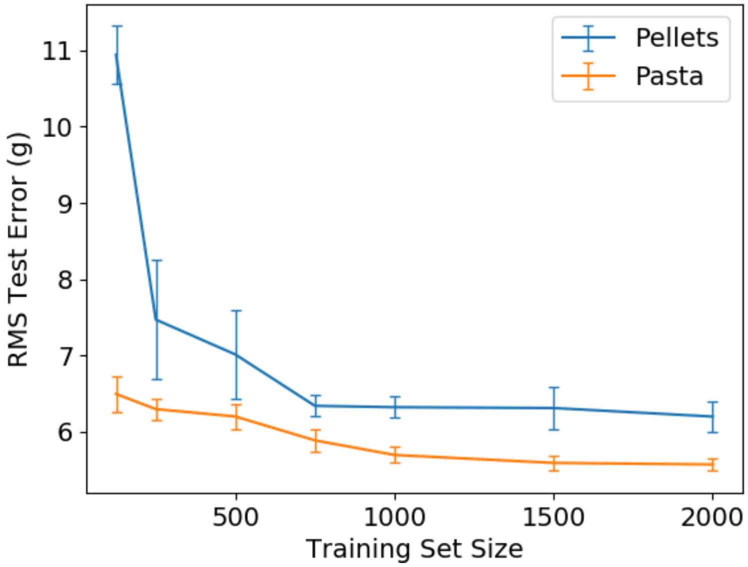


Figure 2.5: The mass prediction accuracy of our model when trained on varying training set sizes.

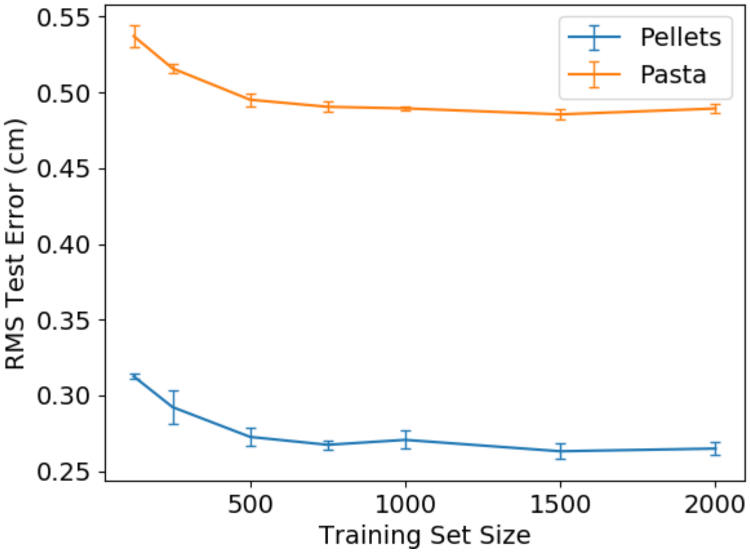


Figure 2.6: The resulting height map prediction accuracy of our model when trained on varying training set sizes.

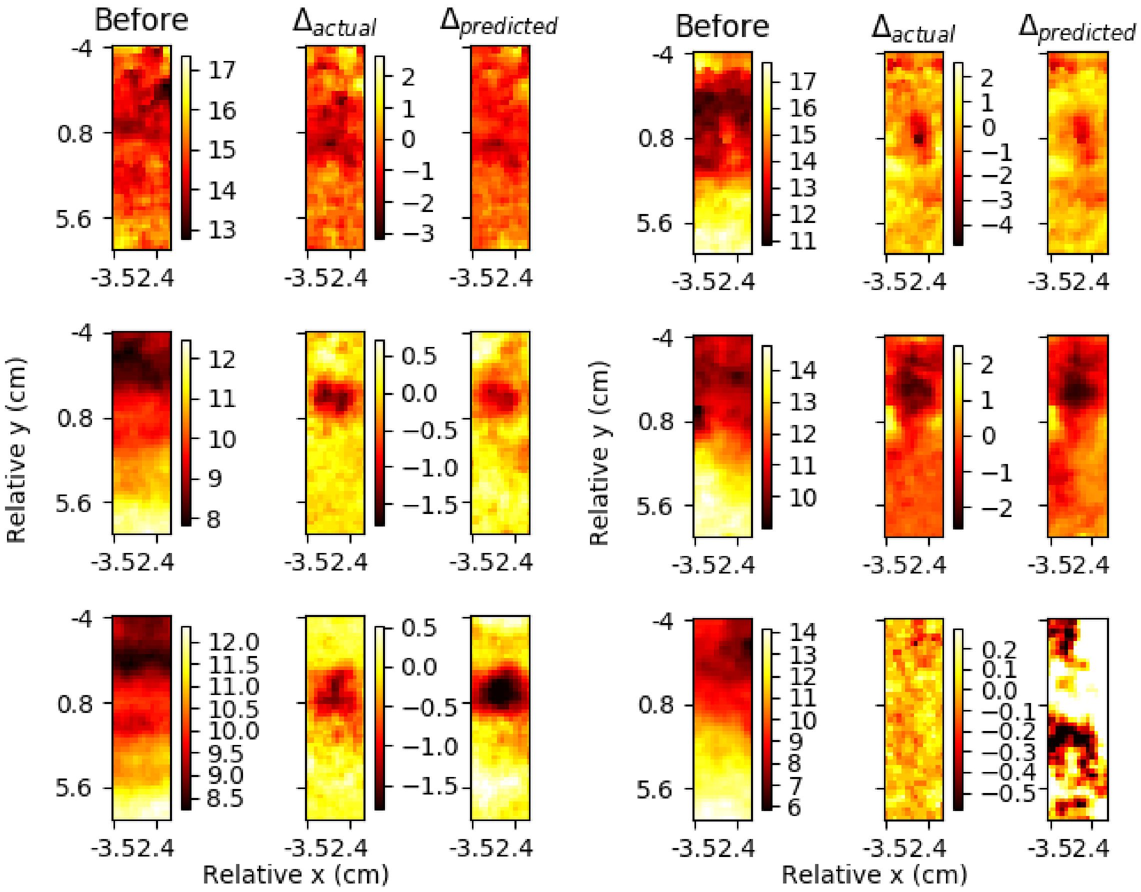


Figure 2.7: Examples of predicted and actual changes in surface heights from scooping actions. The Δ in each subfigure represents the difference between the actual or predicted final height matrix and the initial height matrix. Height dimensions (visualized with the color scale) are in centimeters. **(Left)** Results from the pellets. **(Right)** Results from the pasta.

2.5.3 Control

We also tested the effectiveness of our framework in providing a model for trajectory generation for scooping a desired mass. The task of the experiment was to use the model to select appropriate scooping action parameters, given a height matrix and a desired mass.

For each material, we trained our model on 90% of our dataset and used the remaining 10% as a validation set. We trained through 500 epochs and saved the model at the epoch at which the cross validation error was minimized. A desired mass for the task was then selected uniformly randomly in 2 g increments on the interval from 2 g to 110 g and from 2 g to 64 g for the pellets and pasta, respectively. The x and y value of the scooping action were then sampled uniformly randomly on the same intervals as those used to collect the original dataset. A discretized height matrix was constructed from the surface of the material, centered on this fixed x and y value. This height matrix was used as input to the model, and the remaining four parameters of the scooping action $z_{\mathrm{i}}, z_{\mathrm{f}}, L$, and θ were selected by minimizing the difference between the model's estimate and the desired mass using the cross entropy method, sampling each parameter over the same intervals as those used to collect the dataset. The selected parameters were then used to perform a scooping action, and the actual mass scooped was recorded. The results of 60 trials of the task for each material are shown in Figure 2.8.

2.5.4 Discussion

For the mass prediction task, the root mean squared error (RMSE) of the baseline model estimate across the entire dataset for each material was 43.1 g and 36.4 g for the pellets and pasta, respectively. The RMSE of this baseline model for the resulting surface prediction task was 1.2 cm and 1.6 cm for the pellets and pasta, respectively. For comparison, an even simpler baseline of assuming the surface height map was completely unaffected by

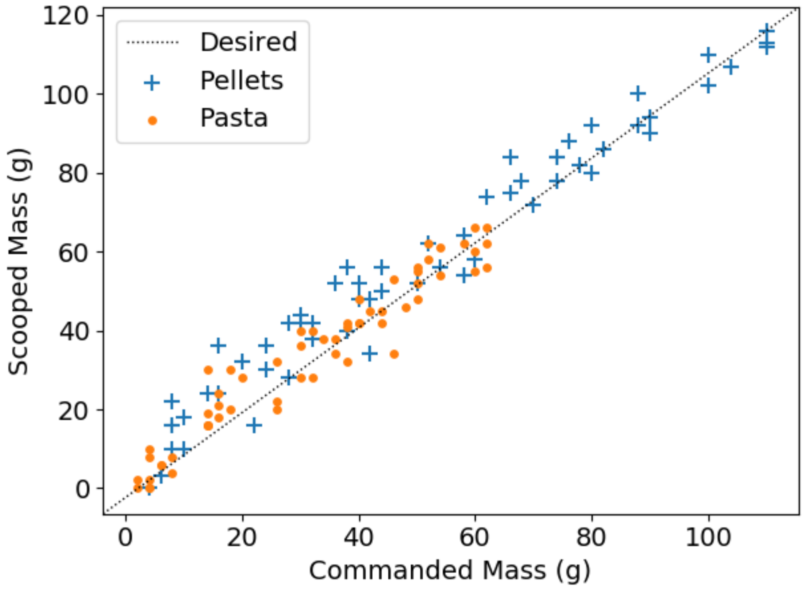


Figure 2.8: The results of 60 trials of using the cross entropy method to select optimal scoop action parameters from a desired mass and a surface height matrix. The mean and standard deviation of the error residuals are 6.4 g and 6.3 g for the pellets and 2.0 g and 5.1 g for the pasta, respectively.

each scooping action yielded an RMSE of 0.75 cm and 0.96 cm for the pellets and pasta, respectively. Thus, in both the mass and shape prediction tasks, our models significantly outperformed the baseline models in each material with only one robot hour of training data. Qualitatively, our model produced shape predictions that were accurate to a high level of detail.

Our model was more accurate in predicting mass for the pasta, especially in the low data regimes. The range of masses in the distribution of scoops for the pellets was higher, as the pellets were nearly three times as dense as the pasta (see Table A.2 in Appendix A.2). Our model predicted the shape changes of the pellets more accurately than those of the pasta. Due to the particles' shape, the pellets have much simpler and more homogeneous dynamics. The irregular shape and large size of the pasta granules makes their motion more difficult to predict to the level of detail required to accurately predict how they will pile after perturbation from a scooping action. However, our model's RMSE of less than

0.55 cm still reflects a strong ability to model the pasta's complex dynamics. Both materials had relatively large and rigid granules with negligible cohesion, but this framework should be evaluated and extended in the future for use on materials with more cohesion, such as wet materials, or materials for which there is more hysteresis, such as fine soil.

To reduce the dimensionality of our inputs, our model assumes a fixed, hand-coded region of interest on the surface of the material, but this region of interest may vary significantly with different materials. A material's major axis length of its granules, angle of repose, etc., could all have a large influence on the size of this region of interest. One future extension could be to learn an attention mechanism to estimate an appropriate size and location of the region of interest for a specific granular material.

In both the mass and surface shape prediction tasks, our framework demonstrated strong sample efficiency given the relatively high dimensionality of the inputs. Note that the models for each material were trained and tested separately. Future work could investigate how well a model is able to generalize to a new material, or to adapt to a new material through few-shot or active learning.

The performance improvements from increasing the training set size attenuated after about 750 examples for each task, representing approximately 6 robot-hours per material. The plateau observed in performance for both materials with more training data may be inevitable with a model which is able to observe only the surface of the material. By only observing the surface of the material, our framework is oblivious to any buried structure or heterogeneity in the material which could affect dynamics of scooping.

For the control task, our framework's performance was commensurate with its performance on the supervised mass estimation task. An interesting extension would be to effectively combine our model with real-time sensor feedback during scooping. This closed-loop control strategy could potentially reduce the controller error beyond our current result by allowing our framework to respond accordingly to any structure in the material that cannot be foreseen from a surface height map. The results we have shown also merit investigating how well our framework can generalize to a new material, or if it can be trained on other materials, then rapidly adapted to a new material after a minimal number of experimental scoops.

2.6 Conclusion

We proposed a novel learning-based approach for modelling the dynamics of granular materials for manipulation tasks. By assuming that the dynamics of granular materials in regions local to their manipulation dominate the relevant dynamics, we reduce the burden of dimensionality on our learning-based model.

We evaluated our framework on a large dataset of scooping two materials with distinct material and dynamic properties. Our framework was able to reliably predict the mass of material that a scooping action would collect, as well as the resulting change in the surface of the materials. Between both materials, the mass prediction and resulting height map estimation tasks were achieved with a worst-case RMSE of 7 g and 0.5 cm, respectively.

The plateau in performance of our architecture suggests that the observation of the height map of the surface of a granular material may not convey enough state to make reliably accurate predictions of dynamics. In the future, the framework could be extended to adjust its estimates using real-time feedback from sensors during the execution of the scooping trajectory. In the next chapter, we present a novel audio feedback mechanism that could be one such feedback mechanism we could incorporate into this framework.

Chapter 3

Learning Audio Feedback for Estimating Amount and Flow of Granular Material

This chapter presents work published in [6].

3.1 Introduction

Sound and structural vibration signals provide a rich source of information for manipulating objects. Humans use this feedback to detect mechanical events and estimate the states of manipulated objects. For example, one may use the sound of a bottle being filled with liquid to estimate how close the bottle is to being full. Similarly, the sound from shaking a near-empty bottle of pills is distinct from the sound of a full bottle, indicating the need to refill the prescription. Experiments have shown that both humans and primates are able to classify distinct types of events (such as whether a dropped glass bottle bounces or breaks [43]), as well as continuous properties of the events (such as the length of a wooden dowel being struck [4]) using only auditory feedback [2, 11].

The ability to sense and process vibrations during manipulation tasks would allow robots to detect and characterize anomalies during manipulation and adapt accordingly. We may be more familiar with vibrations transmitted through air (*i.e.*, sound). Structural vibrations transmit through solid materials and can be sensed through vibrotactile and audio sensors. The cost of collecting and processing vibration feedback is comparatively low relative to other sensor modalities, including vision.

We investigate the use of vibration feedback during the manipulation of granular materials. Granular materials and tasks entailing their manipulation are ubiquitous in both households and industrial environments [8]. We focus on the tasks of pouring and scooping desired amounts of granular materials, exploring whether a robot can use vibration feedback to estimate how much mass it has scooped or how much it has poured.

In the case of pouring a desired amount, we propose to learn models to estimate the amount of material poured based on vibration data collected during the pour. Intuitively, the duration and strength of the vibration should directly correlate with the amount poured. Since pouring is an irreversible process, the amount being poured in any time step is always non-negative, a property that we exploit to provide weak supervision for some of our models.

We evaluate our proposed framework using training data from scooping, shaking, and pouring using our granular materials manipulation setup (shown in Figure 1.1) with a plastic scoop as the end-effector (shown in Figure 3.1). The scoop has a Neewer P-007 contact microphone mounted on it for collecting audio-frequency vibrations throughout the manipulation tasks. For testing the generalization of our frameworks, we use five different granular materials: roasted coffee beans, uncooked Basmati rice, uncooked cellentani pasta, peat top soil, and plastic injection molding beads. Each of these materials has distinct mechanical properties, including different acoustic properties, and is relevant to a different potential application.

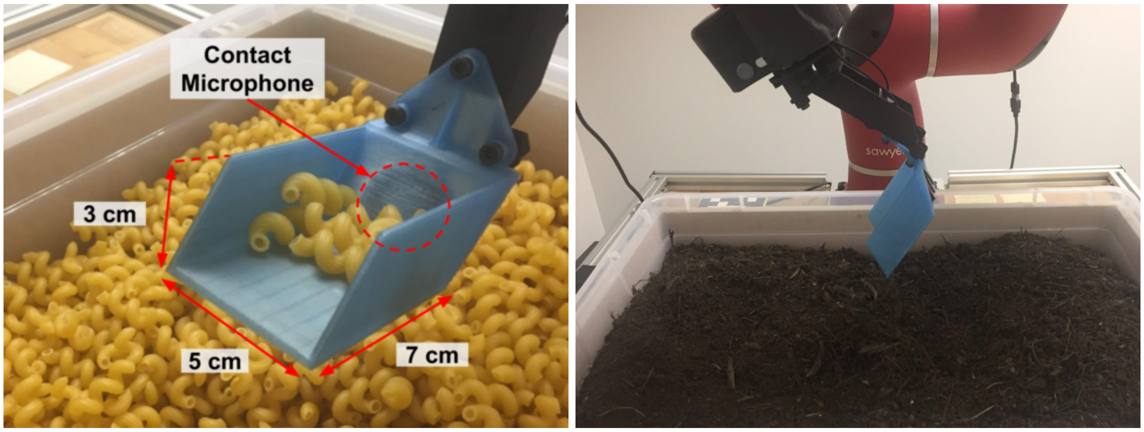


Figure 3.1: Positions of the scoop used for pouring and shaking. **(Left)** The scoop in its *resting position*, prepared to pour or shake cellentani pasta. The silhouette of the contact microphone is visible through the back of the translucent basin of the scoop. **(Right)** The robot with its scoop at its maximum pouring angle, a 60 degree pitch, with the peat top soil below its scoop.

3.2 Related Work

Experiments with humans and primates have investigated the use of auditory feedback to infer characteristics of sound sources. Studies have shown that humans are able to classify sound sources based on sounds emitted during various perturbations [10]. Previous works have also shown that both humans and primates are able to make estimates of metric characteristics of sound sources, such as quantities and geometric dimensions [2, 4, 11, 17].

Various techniques have proven effective for classifying properties of household objects from mechanical vibrations produced by actively manipulating them. Nakamura et al. [24] used audio data, collected from shaking objects with a robot arm, among a multimodal set of features for classifying toys into arbitrary categories. Sinapov et al. [34] used sound signals from a robot actively interacting with different objects (*e.g.*, shaking, pushing, and tapping) to classify and characterize properties of common household objects. Griffith et al. [13] used sound recordings of flowing water striking a container to determine whether it was capturing water or not. Kroemer et al. [16] used a tactile microphone to capture audio-frequency vibrations from a probe stroking materials to learn to classify and cluster the material textures. Saal et al. [28] used recordings from touch sensor arrays on a robot arm's finger tips while shaking a bottle to infer the viscosity of the liquid the bottle contained.

Most similar to our work, Schenck et al. [32] collected features from multiple sensor modalities, including robot joint torques and sound recorded by a microphone, while manipulating containers of granular materials through actions such as dropping and shaking. These features were combined and compared to deduce patterns in matrix completion tasks based on high-level features of objects such as the containers' enclosed materials, colors, and weight ("light," "medium," or "heavy"). Though these frameworks have been effective for their respective classification tasks, our focus is on estimating the amount of material captured or released, a continuous value, from sound recordings. Each of these experiments demonstrates the strength of learning from audio-frequency vibrations to make inferences about physical events and properties of objects in a robot's environment.

With respect to pouring, Yamaguchi and Atkeson [45] used stereo vision to estimate the location and cross section of liquid flow during robotic pouring, using liquids as well as a fine granular material. Schenck and Fox [31] successfully used vision as feedback for learning real-time robotic control of pouring liquids. Though these works demonstrate the strong potential and value of using vision for feedback during robotic pouring, vibrotactile feedback presents unique advantages as an alternative or additional modality of feedback during pouring, *e.g.*, its insensitivity to occlusion and lighting variation.

For materials in containers with constricted openings, Webster and Davies [44] were able to estimate volumes of solid and liquid materials in custom-designed resonator vessels. They actively searched for the resonant frequency of their vessels by applying different frequency vibrations, then used a polynomial regression model based on Helmholtz resonance equations. Our approach does not require specially designed Helmholtz resonator vessels or actively searching for a resonant frequency.

Machine learning techniques for audio-frequency data vary widely based on application and purpose. Many techniques have found converting raw audio to a spectrogram representation to be a powerful tool [12, 13, 29, 34, 46]. Convolutional Neural Networks (CNNs) have been successfully applied to spectrograms in speech recognition and other classification tasks [18, 29, 37]. Other successful approaches to speech recognition from acoustic signals have used recurrent architectures based on Long Short-Term Memory (LSTM) units, which store state in order to learn in the domain of sequential events [36]. Gated Recurrent Units (GRUs) were introduced by Cho et al. [5] as a simpler alternative to LSTM units for recurrent networks. They have been shown to perform well on tasks involving learning from acoustic signals [42], even outperforming LSTMs on some tasks [40].

3.3 Estimation of Amount from Vibratory Feedback

In this section, we describe the different network architectures that we explored for the tasks of estimating amounts and flows of granular materials from audio-frequency vibrations, as well as how the granular material dataset was collected.

3.3.1 Granular Material Manipulation Vibrotactile Dataset

We collected a dataset of audio-frequency vibratory recordings from five different granular materials during shaking and pouring manipulation tasks. To collect this dataset, we used a Rethink Robotics' Sawyer 7-DOF robot arm (shown in Figure 1.1) and designed a 3D printed plastic scoop as its end effector (shown at the left in Figure 3.1). The scoop has a 7 cm long, 5 cm wide, and 3 cm high basin and is equipped with a contact microphone glued to the back outside of its basin for collecting the vibrotactile signal.

We placed a tub containing a granular material in front of the robot. The entire weight of the tub rests on two DYMO M25 scales. The scales each have a measurement resolution of 2 g. The robot scooped random amounts of material from the tub, then alternated between shaking motions and pouring motions, before scooping more material again after the scoop had been emptied. Before each shaking motion and after each pouring motion, the scales measured the mass of the tub to ascertain the mass in the scoop and mass that had been poured, respectively, providing the ground truth mass or flow for each data sample.

The shaking motion was designed to perturb the contents of the scoop enough to make an audible sound, while spilling as little of the scoop's contents as possible. Each shake began with the scoop tilted back to retain material in its *resting position*: a pitch of -15 degrees (shown in the left image of Figure 3.1). Then the robot's joint torques were set to 40% of their maximum torque in an upward and negative-pitch direction for 80 milliseconds before abruptly stopping the robot in its current position. Since the motion was very brief, the majority of the sound occurred well within the first 300 milliseconds of each clip. We therefore truncated each audio clip to its first 500 milliseconds.

For each pouring motion, the pitch of the scoop began at the *resting position* and was then rotated to a random angle between -13 and 60 degrees (shown in the right image of Figure 3.1) using a constant angular velocity sampled uniformly between 12 and 75 deg/sec. Since the angle and velocity of each pour were randomly sampled, the lengths of the audio recordings varied from 0.85 to 6.36 seconds. All of the recordings were zero-padded to 6.4 seconds.

Datasets were collected in this manner with 2,750 shaking and pouring examples for each of five different materials: roasted coffee beans, uncooked Basmati rice, uncooked cellentani pasta, peat top soil, and plastic injection molding beads. These materials were chosen on the basis of their distinct properties, including density, texture, homogeneity, cohesion, and structure, as well as on the basis of their application diversity in both household and industrial settings. Refer to Appendix A.1 for more details about the dataset and Appendix A.2 for more details about each material.

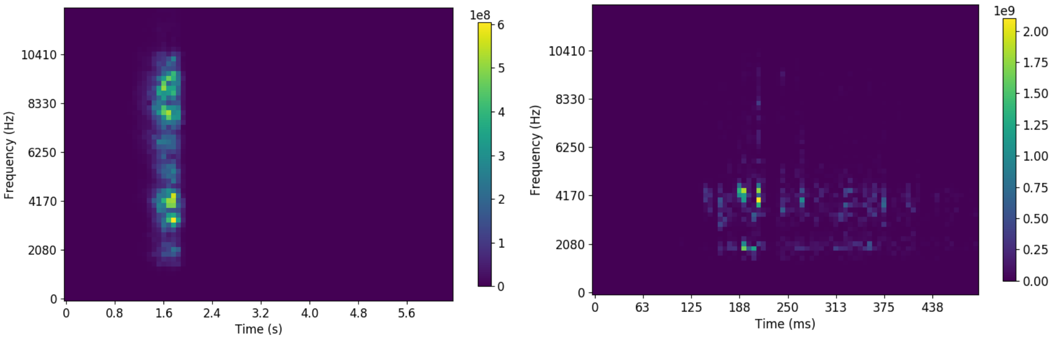


Figure 3.2: Fully preprocessed spectrograms used as input to our frameworks. **(Left)** A spectrogram from pouring 76g of plastic pellets. **(Right)** A spectrogram from shaking 10g of pasta.

3.3.2 Mapping Vibration Signals to Amounts

We compared several different learning frameworks for estimating the amount of material in the scoop, as well as the amount poured, based on the data from the contact microphone. The input to each method was a spectrogram of the audio clip collected during either a shaking or pouring motion. The spectrograms were computed for each audio clip, binning both the time and the frequency at equal intervals from 0 to 6.4 seconds and 0 to 12,500 Hz, respectively. This produced a discretized matrix of power levels within each frequency and time interval, resulting in a 60×80 matrix x for each audio clip, with frequency along the first dimension and time along the second, as shown in Figure 3.2. For this regression task, each method was trained on a mean-squared error loss on the estimation of mass $\phi(x)$ on each element of a subset of examples S, consisting of spectrogram inputs x with ground truth mass values m, as measured by the scales:

$$Loss = \frac{1}{|S|}\sum_{S}(\phi(x) - m)^2 \tag{3.1}$$

For the neural models, we applied this loss during training using minibatch gradient descent, where $|S| = 32$ for each randomly selected batch. For linear regression, we used the full batch of training data to find the analytical regularized least-squares solution.

3.3.3 Linear Regression Baselines

For our linear baseline, we used regularized linear regression. Our input spectrograms have 4800 features, and our dataset has less than 3000 examples per material. For our first linear baseline, we thus used ridge regression to make linear regression tenable. However, naïvely using ridge regression with such a discrepancy in feature dimensionality and training size could be prone to overfitting. Thus, along with this simple linear regression baseline, we devised another linear baseline where we reduced the dimensionality of the input by summing up the input matrices over their time dimensions to produce a vector in frequency space. The resulting features are then proportional to the total energy within each frequency range over the duration of a clip. We then performed ridge regression on these 60 features for our second linear baseline.

3.3.4 Convolutional Neural Network

Convolutional neural networks (CNNs) excel in learning local hierarchical features from structured inputs (*e.g.*, 2D images and 1D audio signals) and have been applied successfully in speech recognition tasks [29]. By sharing parameters in convolution kernels and max pooling over local regions, CNNs are relatively invariant to translations. This invariance and the use of local structure is relevant to spectrograms in which the pouring sounds may occur at different times in each training example.

Our convolutional architecture consisted of a series of convolutional layers with 3x3x8, 4x4x16, and 4x4x32 kernels, respectively. Each convolutional layer was followed by a Rectified Linear Unit (ReLU) activation function and a 2x2 Max Pool with a stride of 2, condensing the output of each layer to the maximum of each non-overlapping 2x2 region. These convolutional layers were followed by two fully-connected layers of size 256, each

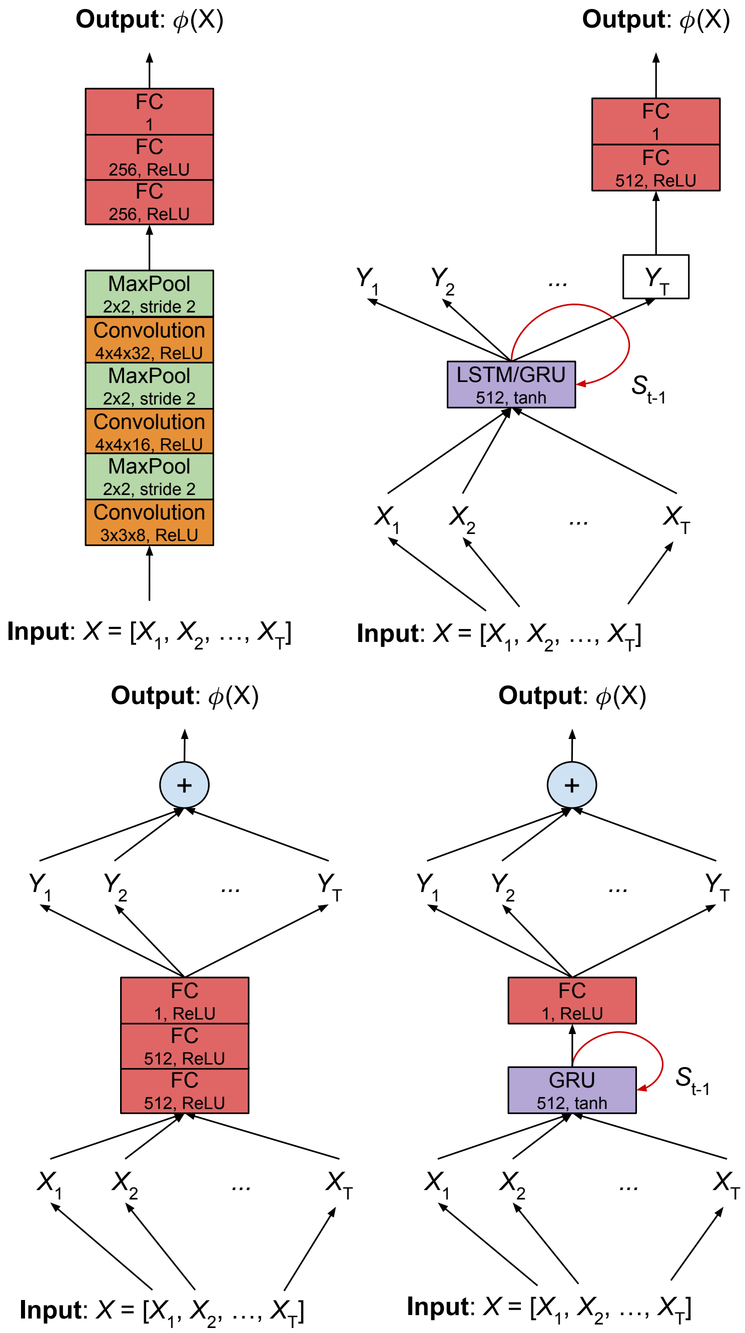


Figure 3.3: Schematics of audio-based learning model architectures. The input X is a spectrogram where the columns $\{X_1, X_2, ...X_T\}$ represent the slices of the spectrogram in its time dimension, *e.g.* X_1 is the first column vector of powers for the different frequencies in the first time bin of the spectrogram. **(Top Left)** Convolutional Neural Network (CNN). **(Top Right)** Recurrent architectures (LSTM/GRU). The only difference between the LSTM and GRU-based architectures was the type of recurrent unit used. **(Bottom Left)** Summing fully-connected network (SumFC). **(Bottom Right)** Summing recurrent network (SumGRU).

with ReLU activations. During training, dropout regularization was applied to the outputs of each of these fully-connected layers, randomly setting each output value to 0 with a probability of 0.5 and multiplying all other values by 2. From the output of the last layer, a linear layer was used to produce the mass estimate $\phi(x)$. See Figure 3.3 for a visualization of this architecture.

3.3.5 Recurrent Networks

Recurrent neural networks are well-suited to tasks involving sequential and temporal data, including audio, which is inherently temporal. Recurrent networks are also well-suited for variable-length inputs, and theoretically can output an estimate of the mass at any point in time. The LSTM unit was designed to mitigate some of the pitfalls of recurrent neural networks by adding differentiable gates to the memory stored by the unit and regulating the propagation of loss gradients through the time dimension. The currently most popular design of LSTM uses three such gates, *i.e.*, an input, an output, and a “forget” gate. The GRU unit was introduced as a simpler alternative to the standard LSTM unit, having only an “update” and a “reset” gate [5]. Diagrams of both units are shown in Figure A.7 of Appendix A.4. Rather than using a gate on the output, the GRU directly outputs its hidden state, reducing its design complexity and the number of parameters that need to be learned. We thus compared the performance of networks based on both LSTM and GRU units.

The recurrent architectures were applied by progressively processing each time slice of frequency power levels through a layer of 512 recurrent units. The final output of this recurrent network was fed to an additional fully-connected layer of 512 units with ReLU activation, followed by a linear layer to produce $\phi(x)$. During training, dropout was applied to the LSTM output, as well as to the output of the intermediate fully-connected layer used in regression. The architecture of the LSTM and GRU networks were identical, the only variation being the type of recurrent units used in the recurrent layer. See Figure 3.3 for a visualization.

3.3.6 Summing Networks

The trained networks should ideally be able to predict the amount of material poured at each time step, such that they can be used for continuous estimation during the pouring process. However, we only provide the total final amount of material poured for the training data. Training the step-wise predictions of the networks is therefore only *weakly* supervised.

In the case of pouring, we can leverage the principle that the amount of material in the scoop is monotonically decreasing, *i.e.*, the amount poured out during any time step must be nonnegative. We use this insight to provide additional structure to the models, constraining them to estimate the mass poured during each time step as a nonnegative value. We then estimate the total poured mass as the cumulative sum of the mass estimate from all previous time steps. In this manner, the framework cannot compensate for overestimates in the material poured by including negative mass flow at a different point in time. We used this principle in both a fully-connected and a recurrent architecture by training each model to estimate a nonnegative mass for each time slice.

The summed fully-connected network (which we call SumFC) applied two 512 unit fully-connected layers, followed by a single unit layer, each with ReLU activations, to each time step of the spectrogram. Its mass estimate $\phi(x)$ was then the sum of the output of this network for each time step. During training, dropout was applied to the output of each layer except the final output layer.

The summed GRU network (which we call SumGRU) followed the same premise as the summed fully-connected network, merely replacing the first two hidden layers with a GRU layer. It consisted of a 512 cell GRU layer followed by a single unit dense layer with ReLU activation, summed over all time steps to yield $\phi(x)$.

In each architecture, since a ReLU activation constrains the output of the final layer to be non-negative for each timestep, the contribution of each timestep to the total sum is non-negative. Visualizations of both architectures are shown in Figure 3.3.

3.4 Evaluation

For the evaluations, we trained each model until the error on a held-out validation set was minimized. We then used the corresponding learned model for the evaluation on a separate held-out test set. Our dataset included many examples of shakes and pours where the scoop was empty (quantified in Table A.1 of Appendix A.1). To ensure that our models were robust on examples that were less trivial, we filtered our validation and test sets to only include examples where the scoop was measured by the scales to be nonempty at the outset. We report the final test error as the average test error from 5 trials on random train-validation-test data splits, unless otherwise specified. Note that each material has a different density and consequently a different distribution of poured masses. Hence, each model is compared with other models on the same material.

3.4.1 Learning for a Single Material

In practice, a robot may only need to manipulate one given granular material, or it may be able to construct and store separate models for each material it needs to manipulate. Thus, we evaluated each of the methods on its ability to train and test on data from the same material. We tested the dataset for each material individually, splitting into 70-15-15 train-validation-test percentages. In addition to our linear regression baselines and proposed models, we also estimated masses from static analysis of the robot's joint torques and present all results in Figure 3.4.

For this task, the recurrent architectures (LSTM, GRU, and SumGRU) consistently performed the best for both the pouring and the shaking estimation, with the exception that the

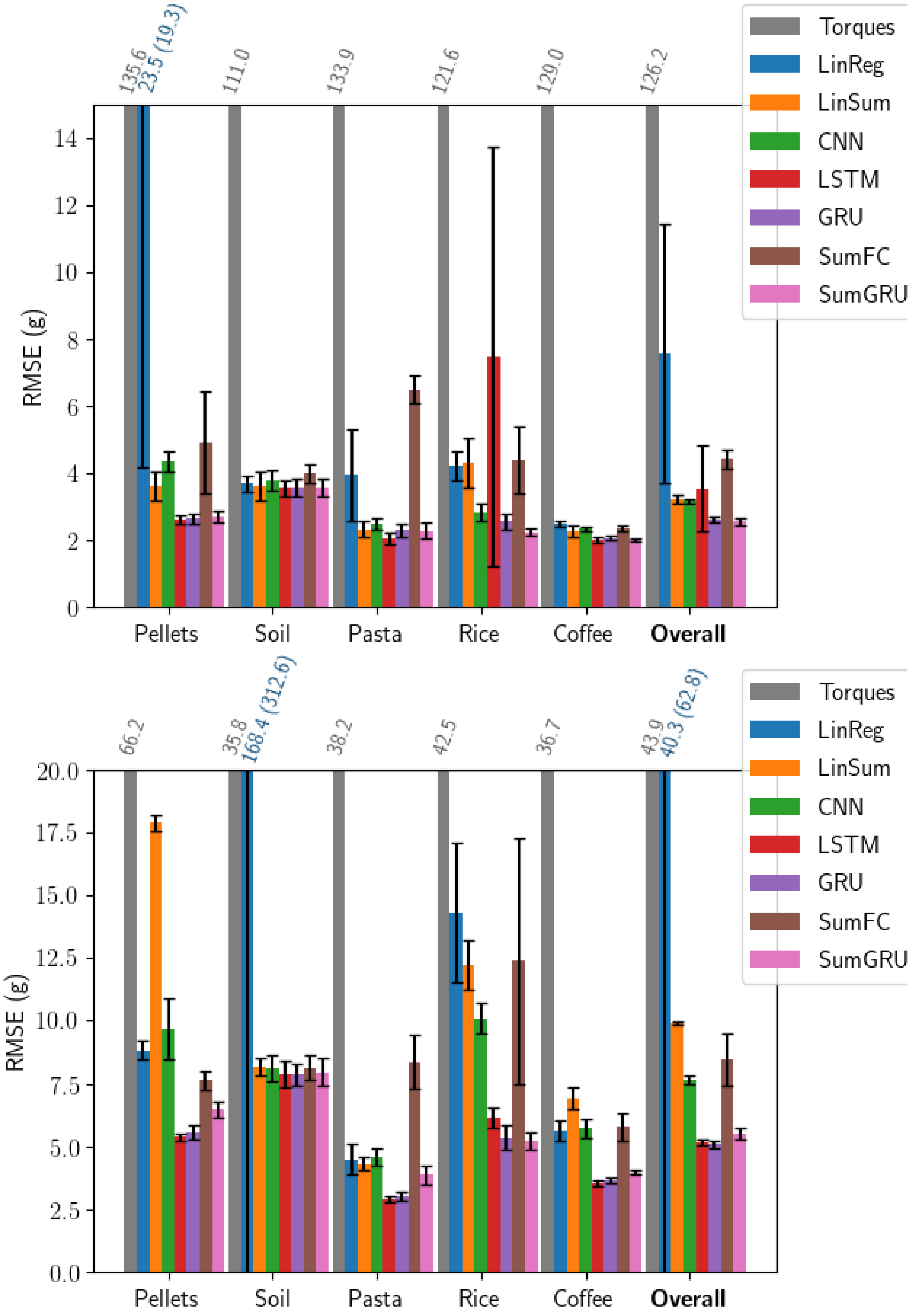


Figure 3.4: Performance of single-material models on estimating mass of each material. Bars that surpass the bounds of the graph have their inter-trial means (and standard deviations) respectively printed above them. **(Top)** Estimating poured mass. **(Bottom)** Estimating shaken mass.

LSTM occasionally was not able to converge during training specifically on the pouring data for rice. We had noticed during the design of these frameworks that the convergence of the LSTM was sensitive to the granularity of the discretization of the spectrogram on its time dimension. It is possible that it may have converged more consistently on the rice data with a coarser discretization. However, the GRU counterpart to the LSTM network consistently converged, perhaps due to its reduced unit complexity. Our models consistently outperformed estimates based on static joint torques, with the exception of the raw regularized linear regression, which was prone to occasionally overfitting.

3.4.2 Learning for All Materials

Rather than using separately trained models for each material it encounters, a robot may benefit from learning one model over multiple materials. This provides the model with more data and may help to avoid overfitting, as the robot must learn features that generalize well across all materials. We thus test each framework's ability to model multiple materials simultaneously and whether it benefits from additional data. To test this approach, we split the data from each material into 70-15-15 train-validation-test percentages, combining all the train and validation sets for the training process, and using the test set from each material separately to test our model's strength for that particular material. These results are shown in Figure 3.5.

Once again, the recurrent architectures performed best. The LSTM apparently benefitted from learning over all the data and extracting useful features, in that it was able to consistently converge and model the rice pouring data effectively. The LSTM architecture also slightly outperformed both GRU architectures on almost all materials, demonstrating its advantage of having more trainable parameters when trained with this larger training set.

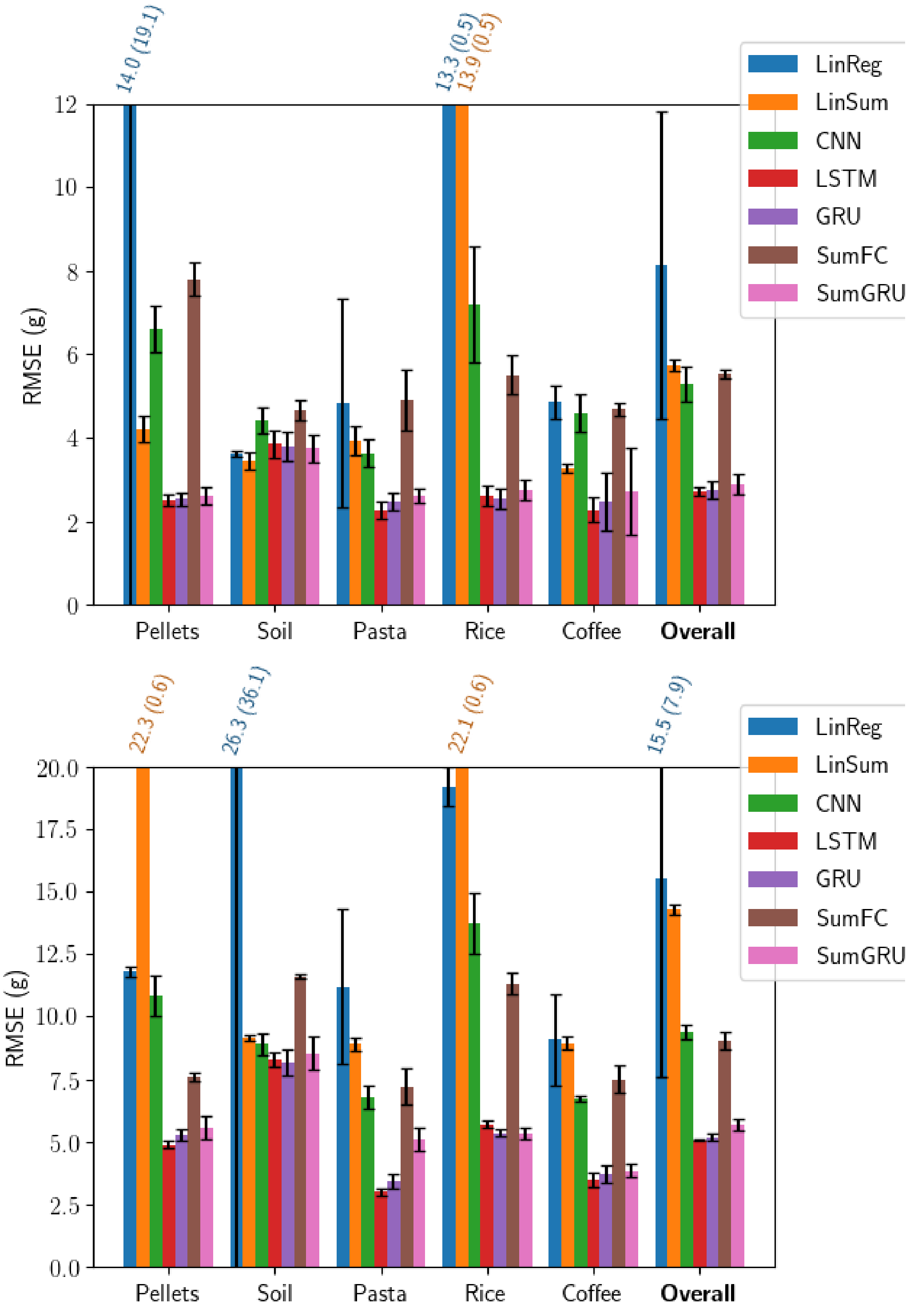


Figure 3.5: Performance of all-material models on estimating mass of each particular material. Bars that surpass the bounds of the graph have their inter-trial means (and standard deviations) respectively printed above them. **(Top)** Estimating poured mass. **(Bottom)** Estimating shaken mass.

3.4.3 Generalizing to New Materials

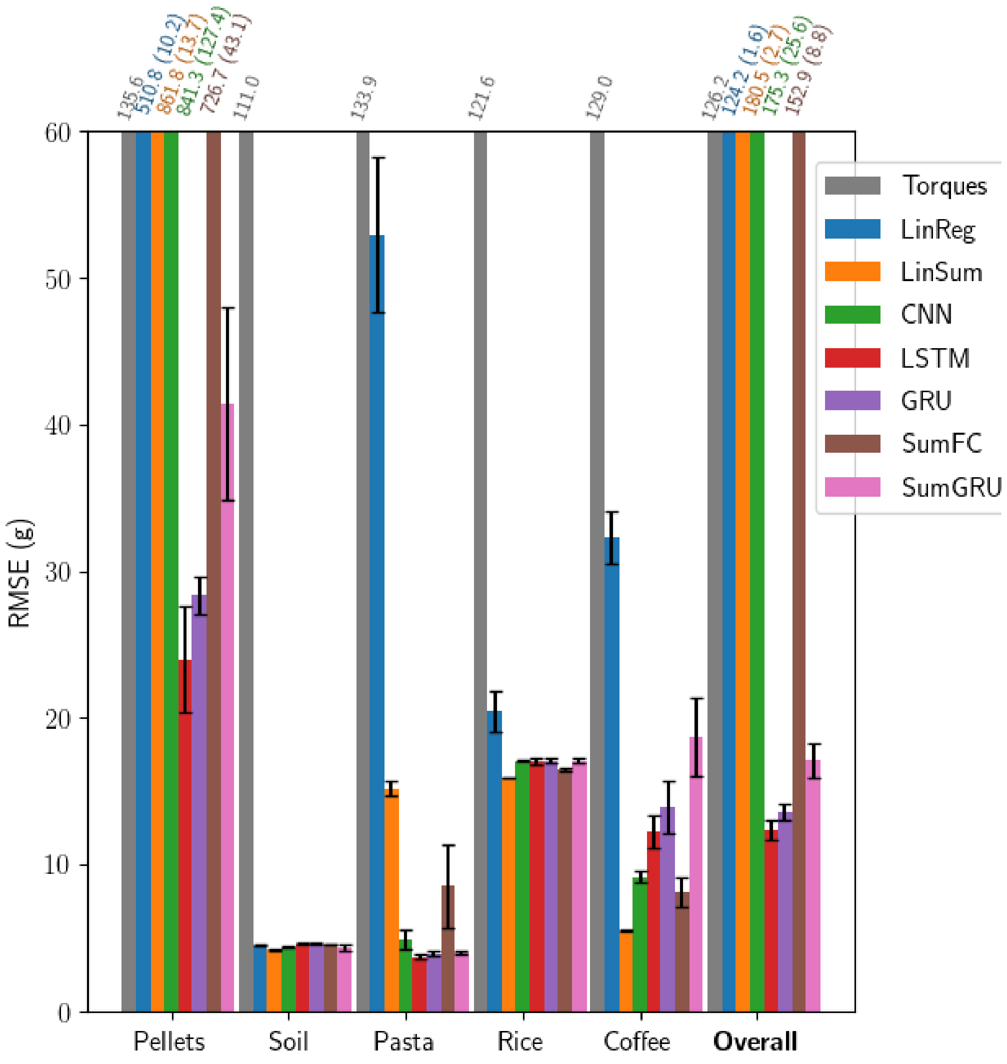


Figure 3.6: Performance of models on estimating *poured* mass of an untrained material, when trained on all other materials. Bars that surpass the bounds of the graph have their inter-trial means (and standard deviations) respectively printed above them.

When a robot encounters a new material, its model should ideally generalize well enough to provide useful feedback on the new material, purely based on what it has previously learned from other materials. In order to test each model's efficacy for generalizing to new materials, we split each dataset into 85-15 training-validation percentages. For each material m, we used a combined training set and validation set from the respective training

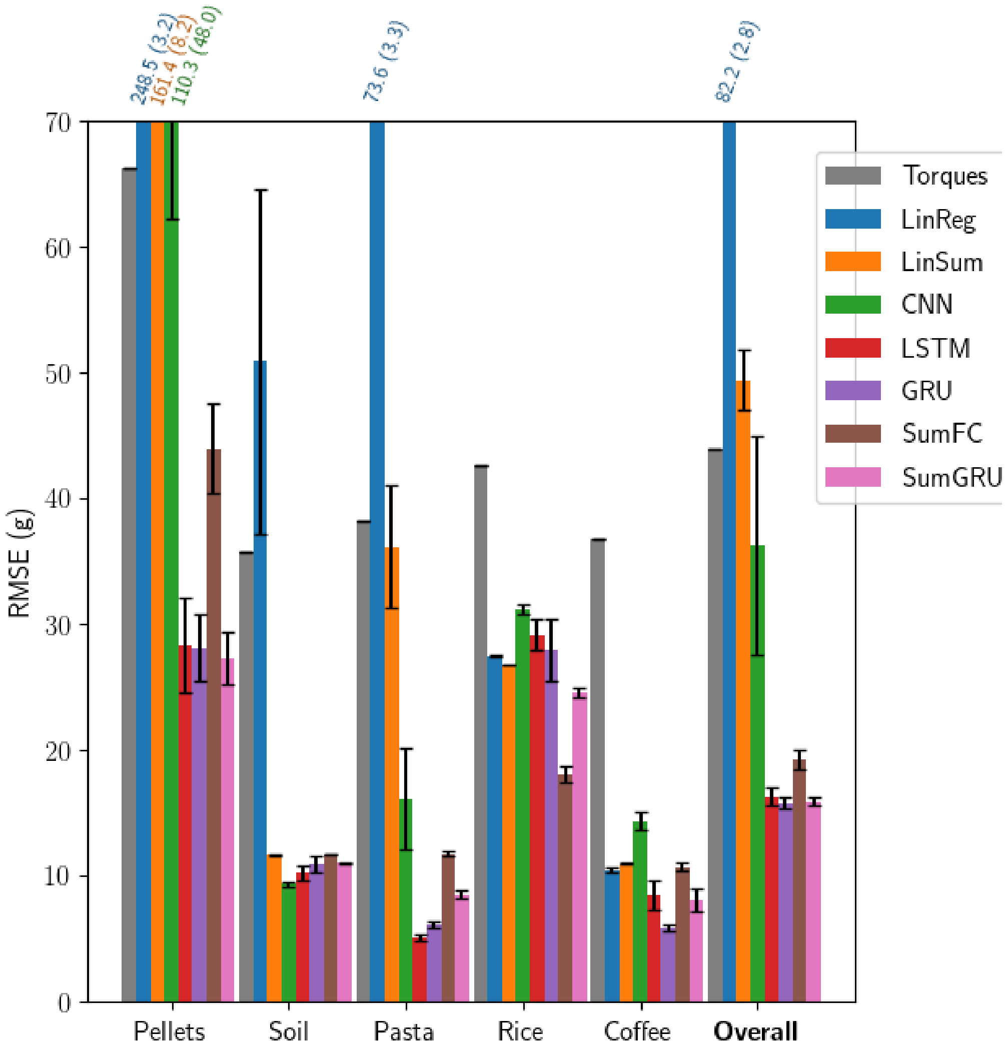


Figure 3.7: Performance of models on estimating *shaken* mass of an untrained material, when trained on all other materials. Bars that surpass the bounds of the graph have their inter-trial means (and standard deviations) respectively printed above them.

and validation sets of all other materials, and tested on the entire dataset of material m for which the scoop was measured to be nonempty. Results from 5 trials of random train-validation splits are shown in for pouring and shaking in Figures 3.6 and 3.7, respectively.

Each neural model performed surprisingly well on the soil and pasta. This may suggest that the neural architectures were able to sufficiently learn features relevant to each of these materials from the data of the other materials, or the discrepancy may simply be due to these datasets' lower means and standard deviations of both their shaken and poured masses relative to the other materials' datasets (see Table A.1 in Appendix A.1). Since the models generalized slightly better to the pasta than coffee datasets, this suggests that both of these factors were likely at play, since the pasta dataset may have higher variance than the coffee, but the pasta is more acoustically similar to the other rigid materials than the coffee is.

Overall the linear, CNN, and SumFC models had high variances on this task, with each of these approaches misestimating the poured mass of the pellets by an order of magnitude more than the rest of the architectures. Specifically for generalizing to pellets, estimates based on static analysis of joint torques outperformed many of the models. The pellets produced a significantly louder noise during pouring, and each of the recurrent networks used $\tanh$ activation functions, which could saturate when confronted with an especially strong input. The SumGRU, however, may have lost some of this benefit by taking the sum of the output from each timestep, sacrificing its ability to correct itself and benefit from saturation. Some normalization of the dataset could have mitigated this issue, but a naïve normalization may not preserve important features of each datapoint, *e.g.*, absolute magnitudes of input features may be too important to disregard. Devising an effective normalization strategy is thus a potential future extension.

3.4.4 Leave-One-Level-Out Cross-validation

To further test the generalization of our models, we tested each model's performance through leave-one-level-out cross-validation. For each material, we separated the dataset into 5 folds based on the percentiles of the masses poured or shaken. The first fold had data examples with masses from the minimum to the 20th percentile, the second fold the 20th to the 40th percentile, etc. Note that since the distributions were skewed, and the resolution of masses was effectively discrete at 2 g intervals, these folds were not necessarily equivalently sized. For each fold, we trained our models on the remainder of the dataset while holding out that fold, then minimized our test error on that fold. We report the average performance over all five folds for each material in Figure 3.8.

On this task, the proposed models performed very similarly on average to how they performed in being trained and tested on all levels of a single material (Section 3.4.1), though they each had much higher variances. Once again, each proposed model outperformed the baseline models overall, with the exception of the LSTM-based model, which was not able to consistently converge on the rice or pellets.

3.4.5 Model Sample Efficiency

In order to test the sample efficiency of each model, we experimented with varying the amount of data on which our models were trained. For each granular material, we held out a test set of 15% of the material's dataset, then trained on different sizes of fixed subsets of the remaining data for that material. For a validation set, we held out the remainder of the material's data which had not been allocated to the training or test sets. We trained until the error on this validation set was minimized, then reported the test error. Results for each model, averaging their performance over all the materials, are shown for pouring and shaking in Figures 3.9 and 3.10, respectively.

The regularized linear regression baseline demonstrated very large variances in the low data regimes. Though the summed linear regression baseline demonstrated much better performance in the low data regimes, both baselines often plateaued in their performance improvements with increases in training set sizes. On the other hand, our proposed models each demonstrated good sample efficiency, even performing well with as few as 125 data samples or approximately one robot-hour of data. Overall, the recurrent models demonstrated the best sample efficiency, with both the GRU-based models outperforming the LSTM-based model. This aligns with empirical results from related work, which found

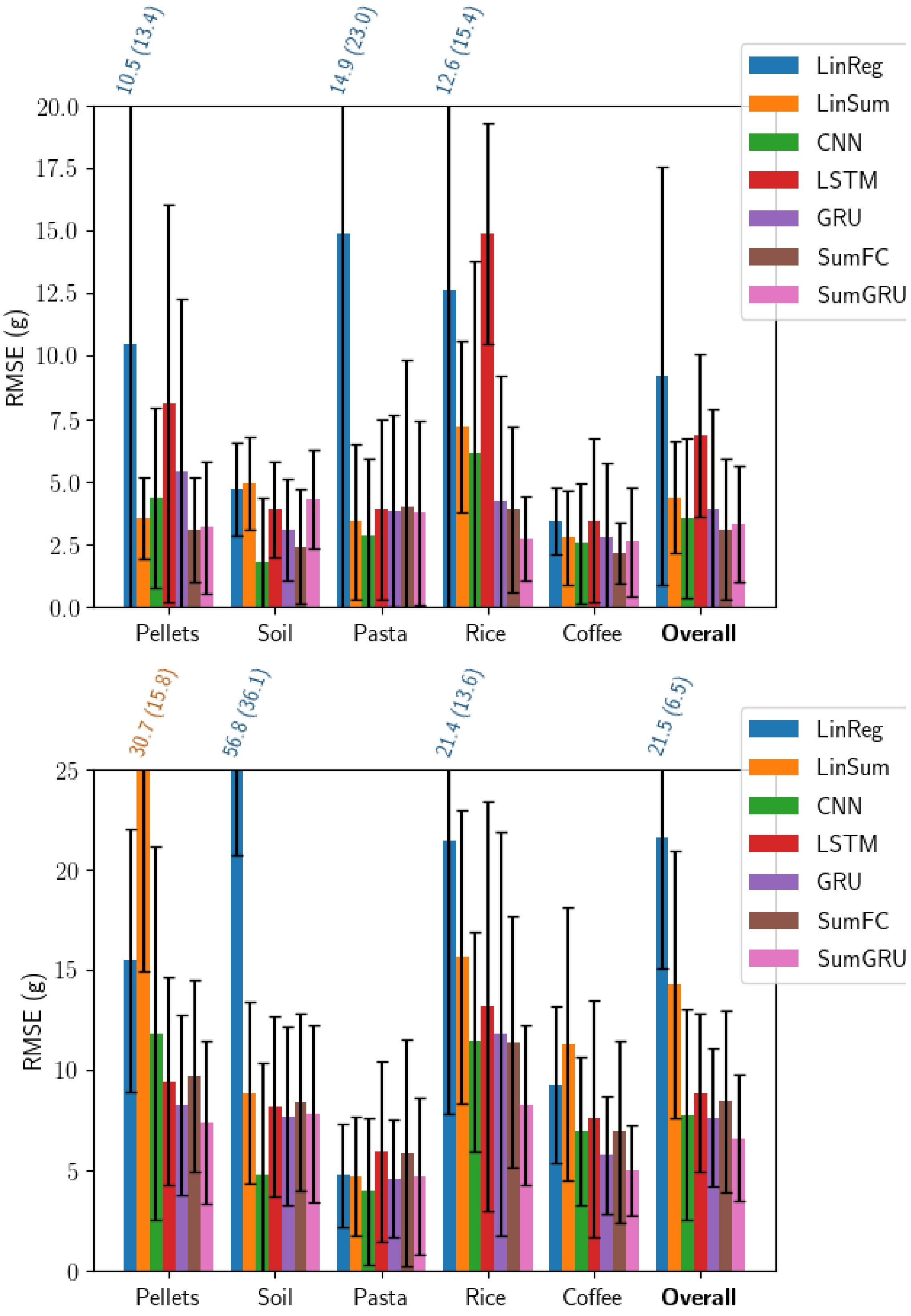


Figure 3.8: Performance of single-material models on estimating mass during leave-one-level-out cross-validation. Bars that surpass the bounds of the graph have their inter-trial means (and standard deviations) respectively printed above them. **(Top)** Estimating poured mass. **(Bottom)** Estimating shaken mass.

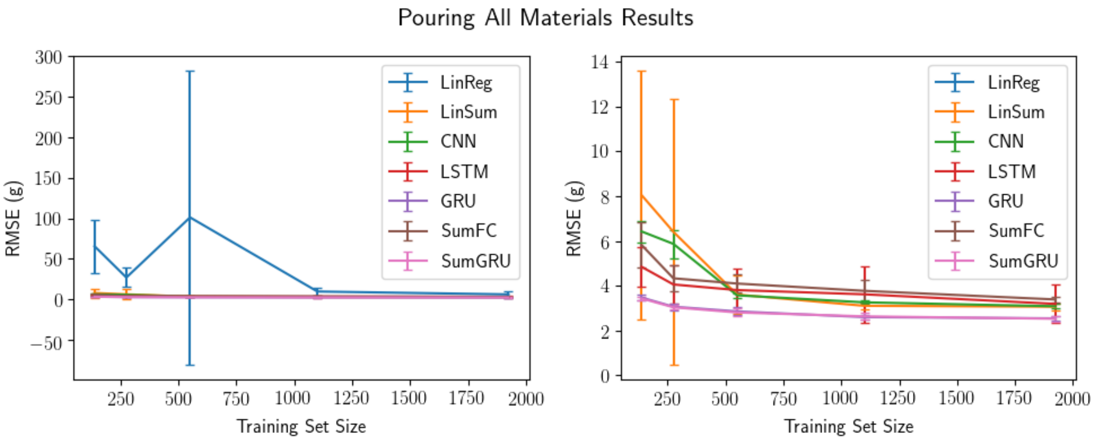


Figure 3.9: Training curves showing variation in performance of models with different sized training sets *averaged over data from pouring all materials*. Results are from averaging over 5 trials. **(Left)** Results from all models. **(Right)** Results from regularized linear regression omitted to show detail of other models' performances.

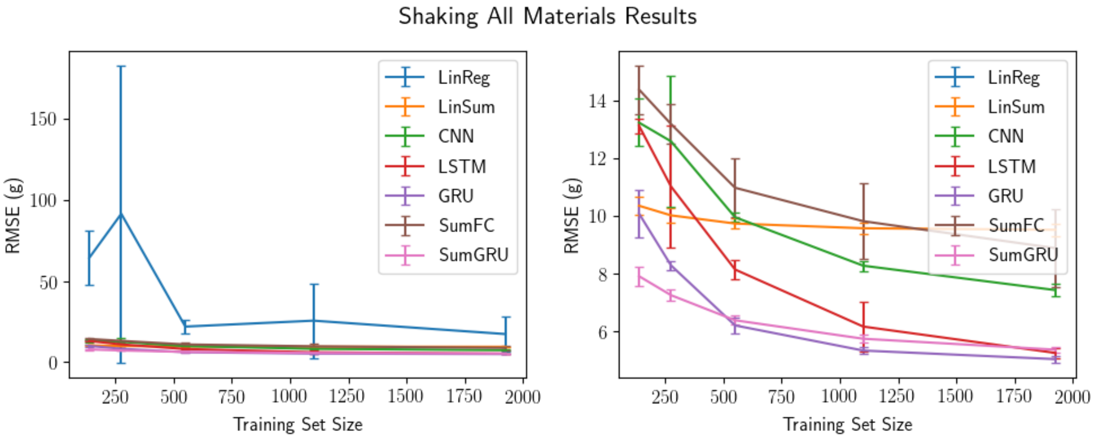


Figure 3.10: Training curves showing variation in performance of models with different sized training sets *averaged over data from shaking all materials*. Results are from averaging over 5 trials. **(Left)** Results from all models. **(Right)** Results from regularized linear regression omitted to show detail of other models' performances.

the GRU to sometimes be more sample efficient, having fewer parameters to learn than the LSTM.

3.4.6 Real-time Control of Pouring

Vibrotactile sensing could potentially provide valuable feedback for robotic manipulation. To demonstrate the robot's ability to use vibrotactile feedback in this setting, we had the robot use a learned model to terminate a pouring skill once it had poured a desired amount.

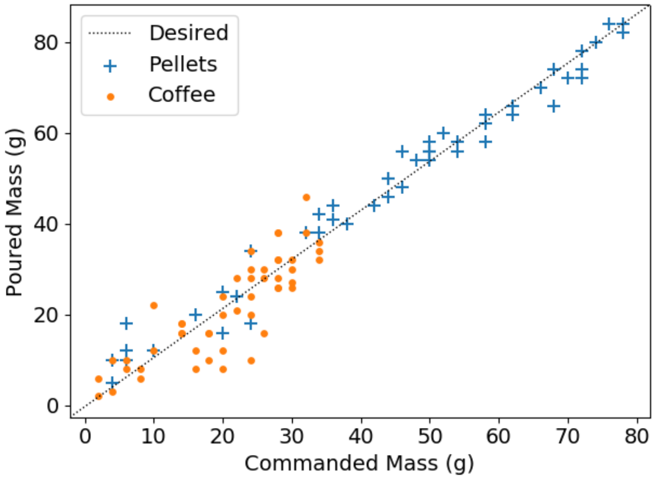


Figure 3.11: Controlling pours of granular material to a desired mass with a SumGRU model, using only tactile audio feedback. The mean and standard deviation of the error residuals are 4.3 g and 3.4 g for the pellets and 0.8 g and 5.8 g for the coffee beans, respectively.

The pours used for our main dataset were too fast to control, and we therefore collected small datasets of spectrograms and resulting masses for pouring both the plastic pellets and the coffee beans, tilting the scoop at 9 degrees/second and terminating each pour at a random duration. We used 800 examples of each material to train separate models based on the SumGRU architecture. These models were used as the feedback mechanism for a basic controller, which terminated the pour immediately upon estimating that the mass poured had reached the commanded mass. Each model made its mass estimates purely based on the current spectrogram of the audio collected so far from the pour. The SumGRU was our most computationally expensive architecture, and the feedback loop was processed at about 20 Hz on a desktop machine with a multi-core processor and an Nvidia Titan

XP graphics card. Of the 50 milliseconds per feedback loop, about 15 milliseconds was spent computing and binning the spectrogram, and the remaining 35 milliseconds was spent passing the spectrogram through the trained SumGRU architecture to generate an estimate of the mass. The results of 50 trials of commanding each controller to pour masses sampled randomly up to the max scoop payload are shown in Figure 3.11.

3.5 Discussion

Each of the models evaluated varied in its relative performance on different tasks and materials. However, patterns that emerged were that the linear baselines, the CNN model, and the SumFC models had the widest variances in their performances. On the other hand, the recurrent architectures consistently were the best performers on almost every task. These architectures are each able to make inferences from relationships between events and features over varying lengths of time. With the inherent temporal and sequential nature of audio data, such relationships may be crucial in extracting the relevant features in this task of mass estimation.

Our cumulative summing networks, SumFC and SumGRU did not show significant performance improvements over the LSTM-based and GRU-based architectures. Thus, our weak supervision of nonnegative pouring flows showed no evidence of significantly improving performance in the architectures for offline estimation. However, they may be useful in real-time mass estimation applications. As shown in Figure 3.12, the standard GRU tends to overshoot in estimating the amount poured during the actual pouring action. By contrast, the SumGRU does not violate the monotonicity constraint, resulting in physically possible flow rate estimates throughout the pouring action. Note that the rise in the estimate of mass for the SumGRU lags behind the most powerful components of the spectra in time, reflecting that its output is not a directly useful real-time estimate. Whereas in

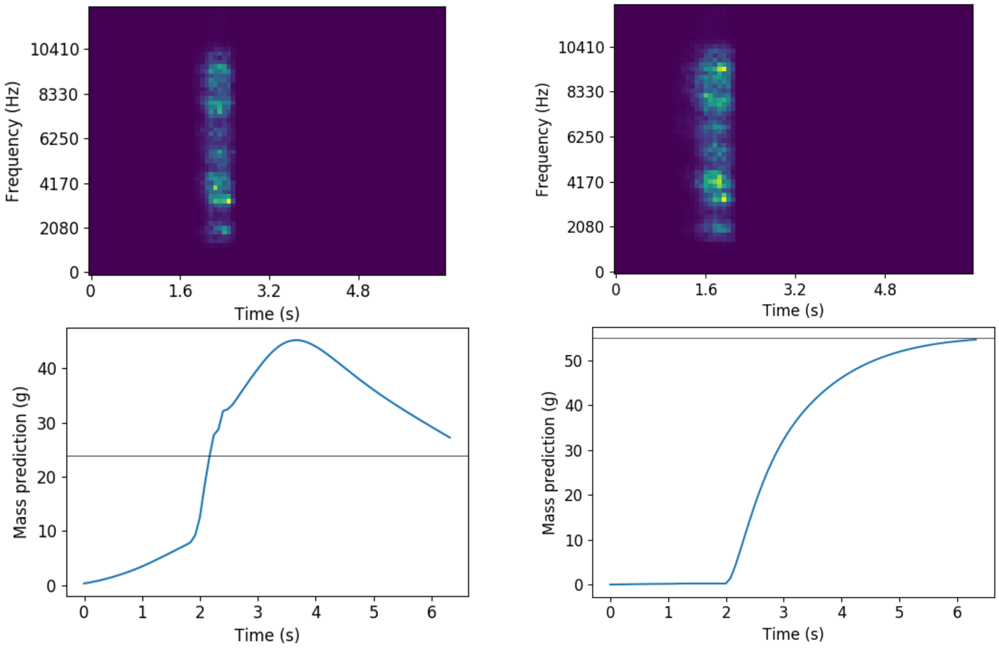


Figure 3.12: Demonstrating the value of the summing architectures for real-time estimation applications, when only training with weak supervision. Each plot shows the spectrogram used as input above the current estimation of mass, as output by the recurrent architecture at each time step. In the mass estimation plots, the ground truth mass is denoted by the horizontal line. **(Left)** The output of the GRU overshoots, then gradually corrects its estimate over time. **(Right)** The SumGRU does not overshoot and approaches the correct estimate monotonically.

our work we collected separate datasets specifically for our example control tasks, future work could investigate training a model from offline data with weak supervision, similar to our initial datasets, to be used in a real time estimation or control task.

We noticed during tuning that the CNN and SumFC models were sensitive to the hyperparameter settings, such as the initial learning rate and learning rate decay schedule. The recurrent architectures were much less sensitive to changes in hyperparameters, though the LSTM struggled to converge when trained only on the rice dataset, as shown in Figure 3.4. We initially had used spectrograms with a finer discretization in the time dimension, but the LSTM often failed to converge with this fine discretization. The performance of our other models was not significantly affected. Future work could investigate using other methods of compressing the time resolution using differentiable methods, such as temporal convolution layers, then combining these with the recurrent layers.

These frameworks each successfully estimated amounts and flows during pouring and shaking. An interesting future extension would be to apply and adapt these methods to estimation of amounts during scooping. In the case of scooping, we expect less correlation of sound with the mass scooped, since the flow of material into the scoop is not necessarily unidirectional. We thus expect that applying these methods directly to estimation of granular material amounts during scooping would be more difficult.

Additional interesting future work could investigate arrangements of multiple microphones, potentially not just contact microphones. Though a signal from a single contact microphone was sufficient for respectable performance on the pouring and shaking tasks, perhaps the signals from multiple microphones of different types could be combined to characterize other events. For example, though a contact microphone can capture the vibrations within the scoop, the signal from a condenser microphone could capture the sound of materials falling from the scoop and striking the surface of the material in the tub. Challenges that may need to be addressed or even leveraged would include robustness to environmental and machine noises, as well as the differences in the speeds of vibrations through different materials and fluids.

3.6 Conclusion

We proposed learning frameworks for estimating amounts and flows of granular material from audio data collected during robotic pouring and shaking tasks. The evaluated methods included state of the art frameworks used in learning for audio signal processing. With an audio signal transformed into a spectrogram, the CNN-based framework was designed to extract hierarchical features from the structure of the spectrogram. The recurrent models, based on LSTM and GRU units, were designed to extract variable-length temporal relationships in the spectrogram. We also proposed a weakly supervised approach to estimating the amount of flow at each time step. The approach exploits the monotonic nature of pouring and applies a nonnegativity constraint to capture the increasing amount of mass poured over time.

We evaluated each approach's effectiveness on a dataset collected from pouring and shaking five distinct granular materials. The frameworks based on recurrent units were consistently the most accurate, with RMSEs near 2.5 g, close to the 2 g measurement resolution from our dataset. They demonstrated strong sample efficiency and were also able to reliably generalize among multiple materials and even to previously unseen materials.

In the future, we will extend the proposed framework to provide continuous low-level feedback control (*e.g.*, servo the tilt angle), and explore additional manipulation tasks (*e.g.*, scooping and cutting). At a more general level, the results of this work show that audio frequency vibrations can be a surprisingly informative sensory modality in robotic tasks, and we plan to further investigate the applications and challenges of using vibratory feedback in robotics going forward.

Appendix A

Audio Feedback

A.1 Dataset Details

As explained in the main manuscript, the robot scooped material, then alternated shaking and pouring the material three times before repeating this process, collecting three shake and three pour recordings per scooping action. This scooping action was randomized in order to vary the initial amount in the scoop and potentially vary the packing and structure of the granular material in the scoop. It was randomized by selecting some of its parameters from random ranges.

The scooping action was performed with parameters as follows (diagrammed in Figure 2.1): the scoop was plunged at a pitch $\pi/5$ from the horizontal at a location specified by the x and y coordinates, to an initial depth z_{i}. The scoop was then drawn through the material on a linear trajectory for length L in the y direction to final depth z_{f}, all while maintaining its pitch. Upon reaching the end of this linear trajectory, the scoop was tilted back about its back edge to a pitch of $\pi/12$ from horizontal in order to retain the material it had scooped before being lifted straight up out of the material. x and y were fixed for all scooping actions, such that each scooping action started roughly in the middle of the tub, offset backward in the y direction to accommodate the length of the scoop. The ranges over which the lengths and depths were sampled were designed to skew toward completely full scoops, since each scoop was followed by three shakes and pours, with $L \in [5, 11]$ cm and $z_{\mathrm{i}}, z_{\mathrm{f}} \in [0.5, 4]$ cm. Throughout its trajectory, the impedance in the x and y direction were set to their maximum values, while the impedance in the z was set to 400 N/m in order to protect the scoop from breaking when it jammed.

After scooping, the robot then moved the scoop slowly to a fixed, constant location above the middle of the tub, high enough above the surface of the material to prevent the scoop from contacting the material in the tub while shaking or pouring. It then performed each of its shakes and pours at this location, resetting to this location after each action.

The scooping and pouring action parameter sampling intervals were designed to produce distributions of masses of poured and shaken material that were as uniform as possible. Significant variations in the materials' properties made this unrealistic. Empty pours or "false" pours (*i.e.* pours in which material was present in the scoop but was retained in the scoop throughout the pouring motion) were especially common. The frequency of such pours is quantified in Table A.1. It was for this reason that, while we trained on all data, we reported the test errors on non-empty pours and shakes to ensure that our reported performances were not bolstered by too many trivial cases of estimating empty shakes or pours from silent recordings. Relevant histograms of masses for each material are shown in Figures A.1 - A.5.

Table A.1: Dataset metadata for each material type.

Material	Empty Pours (%)	Non-empty Pour Mean (g)	Non-empty Pour Std Dev (g)	Non-empty Pour Max (g)	Empty Shakes (%)	Non-empty Shake Mean (g)	Non-empty Shake Std Dev (g)	Non-empty Shake Max (g)
Pellets	17.3	20.0	25.6	78.0	17.3	58.4	29.1	112.0
Soil	27.3	2.4	4.3	32.0	27.5	9.2	7.4	60.0
Pasta	20.2	4.8	7.4	36.0	14.6	12.5	8.7	50.0
Rice	10.6	10.4	14.7	62.0	10.5	30.8	16.5	88.0
Coffee	12.6	6.8	9.4	36.0	12.7	20.6	10.1	52.0

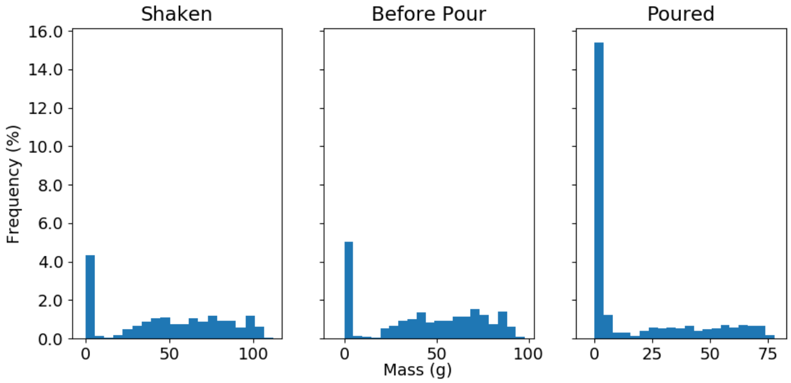


Figure A.1: Distributions of masses for pellets vibrations dataset.

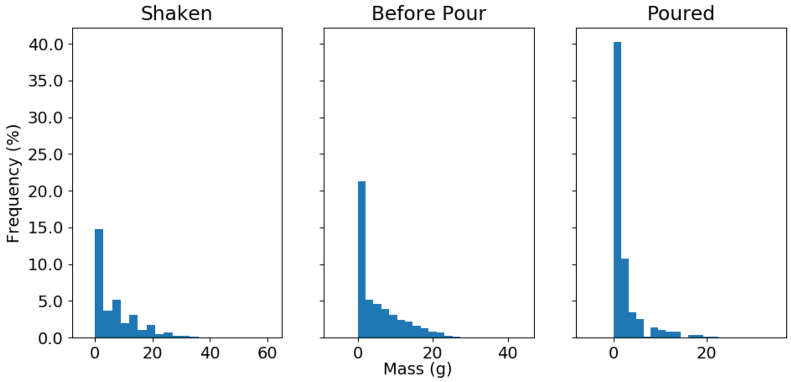


Figure A.2: Distributions of masses for soil vibrations dataset.

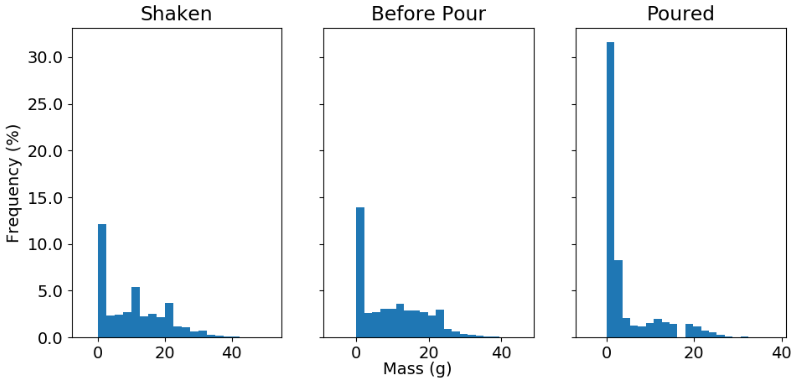


Figure A.3: Distributions of masses for pasta vibrations dataset.

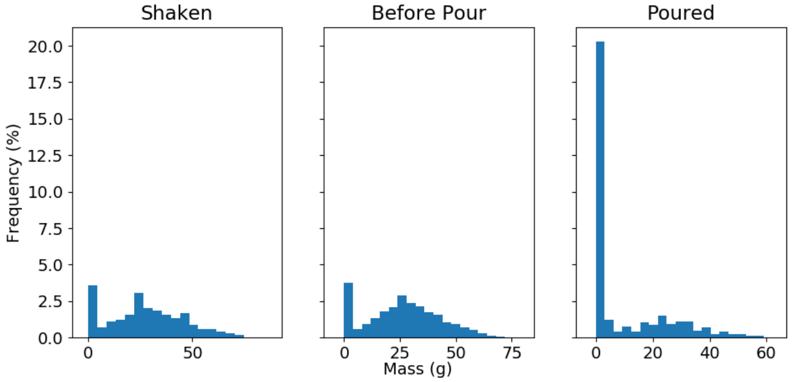


Figure A.4: Distributions of masses for rice vibrations dataset.

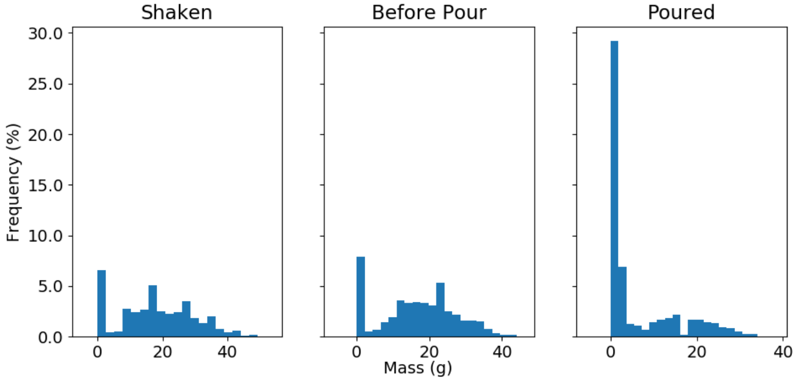


Figure A.5: Distributions of masses for coffee dataset.

A.2 Granular Material Properties

Images of each granular material used in our experiments are shown in Figure 1.3. We also measured some relevant quantitative properties of each material. We selected a random sample of granules of each granular material and measured their mean mass. Note that this was unrealistic for the soil, since the granules of soil were so heterogeneous, ranging from near-microscopic sand and dust particles to 3 cm long wood fragments. We also filled a 1 L beaker of a sample of each material and measured the sample's mass to estimate the packing density of the material. These quantitative measures are shown in Table A.2.

Table A.2: Granular material mass properties. The granules comprising the soil are too heterogeneuous in size to measure a mean single granule mass.

Material	Mean Single Granule Mass (g)	Packing Density (g/cm^3)
Pellets	0.039	0.881
Soil	N/A	0.327
Pasta	1.33	0.355
Rice	0.0154	0.829
Coffee	0.157	0.334

A.3 Contact Microphones and Ambient Noise

Our dataset was collected in an active lab environment, with occasional conversations and cooling fans contributing ambient noise, but perhaps the loudest ambient noise was, in most cases, from the actuation of the robot joints. However, we found that the structural vibrations transmitted through the scoop and recorded by the contact microphone were relatively unaffected by ambient noise, including robot actuation noise. Using a contact microphone directly on the scoop effectively isolated vibrations caused directly by the interaction of the scoop with granular materials. Empirical evidence of this is shown in Figure A.6, as we compare pouring recordings taken simultaneously by our contact microphone adhered to the scoop and a standard lapel microphone pointed forward in the scoop's handle.

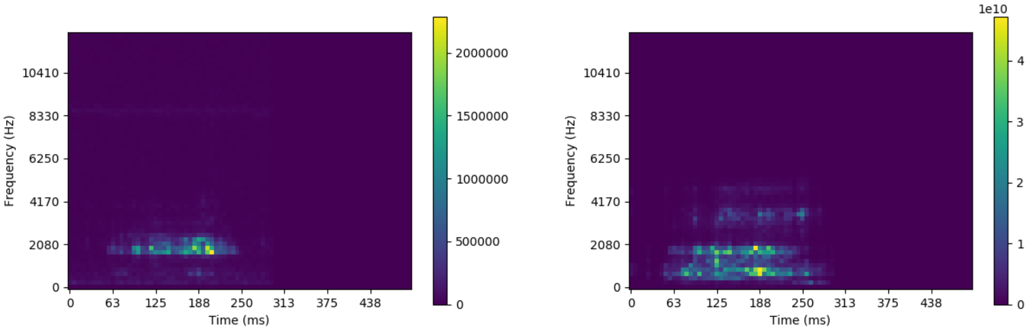
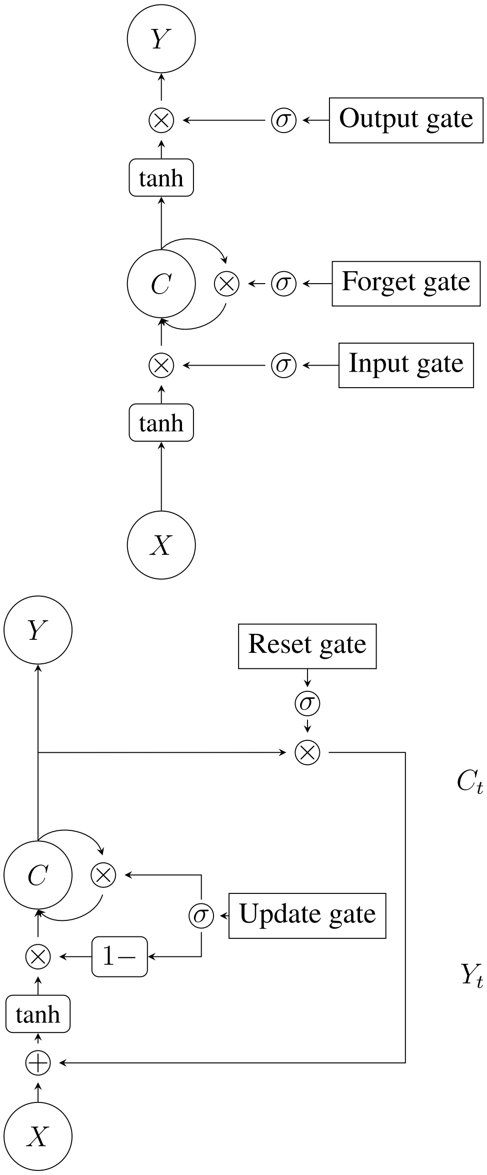


Figure A.6: Spectrograms from different microphones recording simultaneously while the robot is pouring 10 g of soil. **(Left)** Spectrogram from the contact microphone used in our dataset. **(Right)** Spectrogram from a cardioid-profile lapel microphone. Note the more broad range of frequencies represented in the spectrogram of the lapel microphone. These higher frequency peaks are likely due to ambient noise in the lab environment from the robot's actuation, but such vibrations are not as evident in the contact microphone's recording.

A.4 Model Architecture Visualizations

Schematics and equations demonstrating the differences between the LSTM and GRU memory units are shown in Figure A.7. Note that the LSTM unit requires four separate learned weight matrices (W_x, W_f, W_i, and W_o), whereas the GRU unit requires only three (W_x, W_u, and W_r).



$$
\begin{aligned}
C_t &= C_{t-1} \odot \sigma\left(W_f \psi_f\right) \\
&\quad + \tanh\left(W_x X_t\right) \odot \sigma\left(W_i \psi_i\right) \\
Y_t &= \tanh\left(C_t\right) \odot \sigma\left(W_o \psi_o\right)
\end{aligned}
$$

$$
\begin{aligned}
C_t &= C_{t-1} \odot \sigma\left(W_u \psi_u\right) \\
&\quad + \tanh\left(W_x X_t + W_C\left(C_{t-1} \odot \sigma\left(W_r \psi_r\right)\right)\right) \\
&\quad\quad \odot \left(1 - \sigma\left(W_u \psi_u\right)\right) \\
Y_t &= C_t
\end{aligned}
$$

Figure A.7: Comparison of (**Top**) LSTM and (**Bottom**) GRU memory cells. For both diagrams, X is the input, Y is the output, and C is the memory vector. W refers to a learned matrix. ψ refers to an input vector composed by concatenating X and Y_{t-1}. $\odot$ is the Hadamard product. σ is a sigmoid with range $[0, 1]$.

Bibliography

[1] Nathan Bell, Yizhou Yu, and Peter J Mucha. Particle-based simulation of granular materials. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 77–86. ACM, 2005.

[2] Michael J Beran. Quantity judgments of auditory and visual stimuli by chimpanzees (pan troglodytes). *Journal of Experimental Psychology: Animal Behavior Processes*, 38(1):23, 2012.

[3] P. T. De Boer, D.P. Kroese, S. Mannor, and R.Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134, 2002.

[4] Claudia Carello, Krista L Anderson, and Andrew J Kunkler-Peck. Perception of object length by sound. *Psychological science*, 9(3):211–214, 1998.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[6] Samuel Clarke, Travers Rhodes, Christopher G. Atkeson, and Oliver Kroemer. Learning audio feedback for estimating amount and flow of granular material. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd*

Conference on Robot Learning, volume 87 of *Proceedings of Machine Learning Research*, pages 529–550. PMLR, 29–31 Oct 2018. URL `http://proceedings.mlr.press/v87/clarke18a.html`.

[7] Paul W Cleary and Mark L Sawley. Dem modelling of industrial granular flows: 3d case studies and the effect of particle shape on hopper discharge. *Applied Mathematical Modelling*, 26(2):89–111, 2002.

[8] Paul W Cleary and Mark L Sawley. DEM modelling of industrial granular flows: 3D case studies and the effect of particle shape on hopper discharge. *Applied Mathematical Modelling*, 26(2):89–111, 2002.

[9] Dhiraj Gandhi, Lerrel Pinto, and Abhinav Gupta. Learning to fly by crashing. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 3948–3955. IEEE, 2017.

[10] Bruno L Giordano and Stephen McAdams. Material identification of real impact sounds: Effects of size variation in steel, glass, wood, and plexiglass plates. *The Journal of the Acoustical Society of America*, 119(2):1171–1181, 2006.

[11] Massimo Grassi. Do we hear size or sound? Balls dropped on plates. *Perception & Psychophysics*, 67(2):274–284, Feb 2005. ISSN 1532-5962. doi: 10.3758/BF03206491. URL `https://doi.org/10.3758/BF03206491`.

[12] A. Graves, A.-r. Mohamed, and G. Hinton. Speech Recognition with Deep Recurrent Neural Networks. *ArXiv e-prints*, March 2013.

[13] Shane Griffith, Vladimir Sukhoy, Todd Wegter, and Alexander Stoytchev. Object categorization in the sink: Learning behavior–grounded object categories with water.

In *Proceedings of the 2012 ICRA Workshop on Semantic Perception, Mapping and Exploration*. Citeseer, 2012.

[14] Ning Guo and Jidong Zhao. A coupled fem/dem approach for hierarchical multiscale modelling of granular media. *International Journal for Numerical Methods in Engineering*, 99(11):789–818, 2014.

[15] Youssef MA Hashash, Séverine Levasseur, Abdolreza Osouli, Richard Finno, and Yann Malecot. Comparison of two inverse analysis techniques for learning deep excavation response. *Computers and geotechnics*, 37(3):323–333, 2010.

[16] Oliver Kroemer, Christoph H Lampert, and Jan Peters. Learning dynamic tactile sensing with robust vision-based training. *IEEE transactions on robotics*, 27(3):545–557, 2011.

[17] Stephen Lakatos, Stephen McAdams, and René Caussé. The representation of auditory source characteristics: Simple geometric form. *Perception & Psychophysics*, 59(8):1180–1190, Dec 1997. ISSN 1532-5962. doi: 10.3758/BF03214206. URL `https://doi.org/10.3758/BF03214206`.

[18] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, pages 1096–1104, 2009.

[19] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.

[20] Chen Li, Paul B Umbanhowar, Haldun Komsuoglu, Daniel E Koditschek, and Daniel I

Goldman. Sensitive dependence of the motion of a legged robot on granular media. *Proceedings of the National Academy of Sciences*, 106(9):3029–3034, 2009.

[21] Chen Li, Tingnan Zhang, and Daniel I Goldman. A terradynamics of legged locomotion on granular media. *science*, 339(6126):1408–1412, 2013.

[22] Xiaoshan Lin and T-T Ng. A three-dimensional discrete element model using arrays of ellipsoids. *Geotechnique*, 47(2):319–329, 1997.

[23] Rudranarayan M Mukherjee and Ryan Houlihan. Massively parallel granular media modeling of robot-terrain interactions. In *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 71–78. American Society of Mechanical Engineers, 2012.

[24] Tomoaki Nakamura, Takayuki Nagai, and Naoto Iwahashi. Multimodal object categorization by a robot. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2415–2420, Oct 2007. doi: 10.1109/IROS.2007.4399634.

[25] Rahul Narain, Abhinav Golas, and Ming C Lin. Free-flowing granular materials with two-way solid coupling. *ACM Transactions on Graphics (TOG)*, 29(6):173, 2010.

[26] Robert Paolini, Alberto Rodriguez, Siddhartha S Srinivasa, and Matthew T Mason. A data-driven statistical framework for post-grasp manipulation. *The International Journal of Robotics Research*, 33(4):600–615, 2014.

[27] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 3406–3413. IEEE, 2016.

[28] Hannes P Saal, Jo-Anne Ting, and Sethu Vijayakumar. Active estimation of object

dynamics parameters with tactile sensors. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 916–921. IEEE, 2010.

[29] Tara N Sainath and Carolina Parada. Convolutional neural networks for small-footprint keyword spotting. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[30] Shigeru Sarata, Hisashi Osumi, Yoshihiro Kawai, and Fumiaki Tomita. Trajectory arrangement based on resistance force and shape of pile at scooping motion. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 3488–3493. IEEE, 2004.

[31] Connor Schenck and Dieter Fox. Visual closed-loop control for pouring liquids. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2629–2636. IEEE, 2017.

[32] Connor Schenck, Jivko Sinapov, David Johnston, and Alexander Stoytchev. Which object fits best? solving matrix completion tasks with a humanoid robot. *IEEE Transactions on Autonomous Mental Development*, 6(3):226–240, 2014.

[33] Connor Schenck, Jonathan Tompson, Dieter Fox, and Sergey Levine. Learning robotic manipulation of granular media. *CoRR*, abs/1709.02833, 2017. URL `http://arxiv.org/abs/1709.02833`.

[34] Jivko Sinapov, Mark Wiemer, and Alexander Stoytchev. Interactive learning of the acoustic properties of household objects. In *ICRA*, pages 2518–2524. IEEE, 2009. URL `http://dblp.uni-trier.de/db/conf/icra/icra2009.html#SinapovWS09`.

[35] S. Singh. Learning to predict resistive forces during robotic excavation. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 2, pages 2102–2107 vol.2, May 1995. doi: 10.1109/ROBOT.1995.526025.

[36] Hagen Soltau, Hank Liao, and Hasim Sak. Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition. *arXiv preprint arXiv:1610.09975*, 2016.

[37] Elias Sprengel, Martin Jaggi, Yannic Kilcher, and Thomas Hofmann. Audio based bird species identification using deep learning techniques. In *LifeCLEF 2016*, number EPFL-CONF-229232, pages 547–559, 2016.

[38] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[39] Robert W Sumner, James F O'Brien, and Jessica K Hodgins. Animating sand, mud, and snow. In *Computer Graphics Forum*, volume 18, pages 17–26. Wiley Online Library, 1999.

[40] Yaodong Tang, Yuchen Huang, Zhiyong Wu, Helen Meng, Mingxing Xu, and Lianhong Cai. Question detection from acoustic features using recurrent neural network with gated recurrent unit. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 6125–6129. IEEE, 2016.

[41] John M Ting, Mahmood Khwaja, Larry R Meachum, and Jeffrey D Rowell. An ellipse-based discrete element model for granular materials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 17(9):603–623, 1993.

[42] Toan H Vu and Jia-Ching Wang. Acoustic scene and event recognition using recurrent

neural networks. *Detection and Classification of Acoustic Scenes and Events*, 2016, 2016.

[43] William H Warren and Robert R Verbrugge. Auditory perception of breaking and bouncing events: A case study in ecological acoustics. *Journal of Experimental Psychology: Human perception and performance*, 10(5):704, 1984.

[44] Emile S Webster and Clive E Davies. The use of Helmholtz resonance for measuring the volume of liquids and solids. *Sensors*, 10(12):10663–10672, 2010.

[45] Akihiko Yamaguchi and Christopher G. Atkeson. Stereo vision of liquid and particle flow for robot pouring. In *16th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2016, Cancun, Mexico, November 15-17, 2016*, pages 1173–1180, 2016. doi: 10.1109/HUMANOIDS.2016.7803419. URL `https://doi.org/10.1109/HUMANOIDS.2016.7803419`.

[46] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. L. Yoshua Bengio, and A. Courville. Towards end-to-end speech recognition with deep convolutional neural networks. *ArXiv e-prints*, January 2017.

[47] Yongning Zhu and Robert Bridson. Animating sand as a fluid. *ACM Transactions on Graphics (TOG)*, 24(3):965–972, 2005.