Learning Audio Feedback for Estimating Amount and Flow of Granular Material

Samuel Clarke, Travers Rhodes, Christopher G. Atkeson, and Oliver Kroemer
Robotics Institute
Carnegie Mellon University
United States
{sclarke1, traversr, cga, okroemer}@cs.cmu.edu

Abstract: 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 2.8g across the five materials for pouring. We also demonstrate how the learned networks can be used to pour a desired amount of material.

Keywords: vibrotactile sensing, mass estimation, deformable materials

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 (*e.g.*, whether a dropped glass bottle bounces or breaks [1]), as well as continuous properties of the events (*e.g.*, the length of a wooden dowel being struck [2]), using only auditory feedback [3, 4].

The ability to sense and process vibrations during manipulation tasks would allow robots to detect and characterize anomalies during manipulation and adapt accordingly. Whereas 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 (*e.g.*, vision).

In this paper, 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 [5]. 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. Also, 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 data from scooping, shaking, and pouring using a 7-DOF Sawyer robot arm with a plastic scoop as the end-effector (shown in Figure 1). The scoop has a Neewer P-007 contact microphone mounted on it for collecting audio-frequency vibrations

2nd Conference on Robot Learning (CoRL 2018), Zürich, Switzerland.

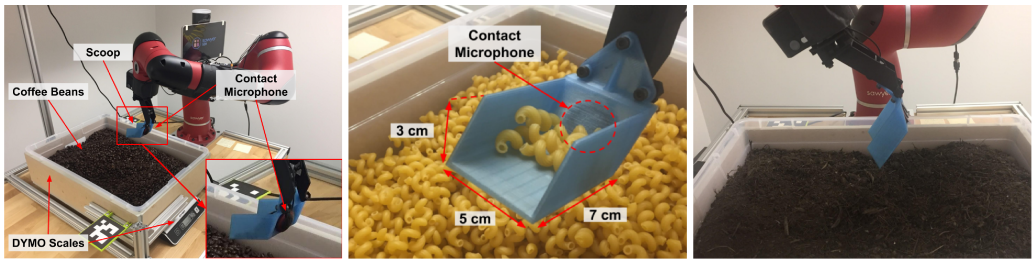

Figure 1: **(Left)** The robotic granular material manipulation setup used to collect a dataset. **(Center)** The scoop in its *resting position*, prepared to pour or shake cellentani pasta. The silhouette of the contact microphone is visible through the back of the translucent basin of the scoop. **(Right)** The robot with its scoop at its maximum pouring angle, a 60 degree pitch, with the peat top soil below its scoop.

throughout the manipulation tasks. For testing the generalization of our frameworks, we use five different granular materials: roasted coffee beans, raw Basmati rice, raw 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.

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 [6]. 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, 3, 4, 7].

Various techniques have proven effective for classifying properties of household objects from mechanical vibrations produced by actively manipulating them. Nakamura et al. [8] 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. [9] 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. [10] used sound recordings of flowing water striking a container to determine whether it was capturing water or not. Kroemer et al. [11] 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. [12] 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. [13] 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 [14] 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 [15] 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 modality of feedback during pouring, *e.g.*, its insensitivity to occlusion and lighting variation.

For materials in containers with constricted openings, Webster and Davies [16] 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 [9, 10, 17, 18, 19]. Convolutional Neural Networks (CNNs) have been successfully applied to spectrograms in speech recognition and other classification tasks [17, 20, 21]. 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 [22]. Gated Recurrent Units (GRUs) were introduced by Cho et al. [23] as a simpler alternative to LSTM units for recurrent networks. They have been shown to perform well on tasks involving learning from acoustic signals [24], even outperforming LSTMs on some tasks [25].

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.

3.1 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 and designed a 3D printed plastic scoop as its end effector, as shown in Figure 1. The scoop has a 7 cm long, 5 cm wide, and 3 cm high basin and is equipped with a contact microphone adhered 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 content 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 center image of Figure 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 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, raw Basmati rice, raw 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 for more details about the dataset and Appendix B for more details about each material.

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

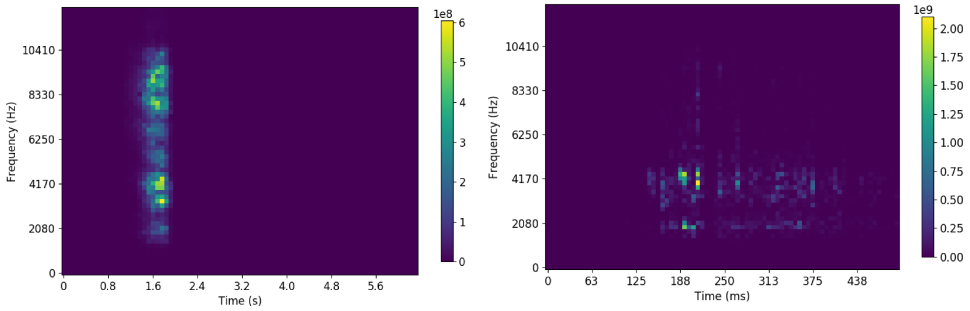

Figure 2: Fully preprocessed spectrograms used as input to our frameworks. **(Left)** A spectrogram from pouring 76g of plastic pellets. **(Right)** A spectrogram from shaking 10g of pasta.

Figure 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:

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

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 Linear Regression Baselines

For our linear baseline, we used regularized linear regression. Our input spectrograms correspond to 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.

3.4 Convolutional Neural Network

Convolutional neural networks (CNNs) excel in learning local hierarchical features on structured inputs (*e.g.*, 2D images and 1D audio signals) and have been applied successfully in speech recognition tasks [17]. 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 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 .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 16 in Appendix D for a visualization of this architecture.

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 [23]. Diagrams of both units are shown in Figure 15 of Appendix D. 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 feeding each time slice of frequency power levels to 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 16 of in Appendix D for a visualization.

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 16 of Appendix D.

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 1 of Appendix A). 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. In addition to the evaluations presented here, we show more results on our frameworks' generalization in Appendix E and sample efficiency in Appendix F.

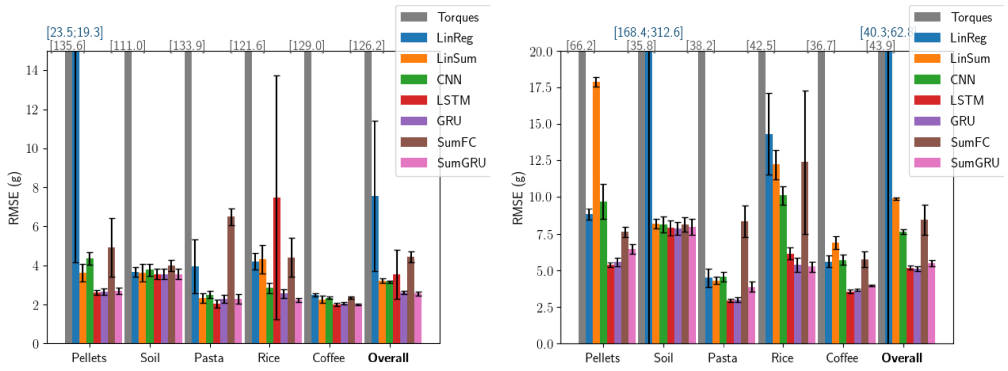

Figure 3: Performance of single-material models on estimating mass of each material. Bars that surpass the bounds of the graph have their inter-trial means and standard deviations respectively printed above them. **(Left)** Estimating poured mass. **(Right)** Estimating shaken mass.

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 their 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.

For this task, the recurrent architectures consistently performed the best for both the pouring and the shaking estimation, with the exception that the 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 complexity. Our models consistently outperformed estimates based on static torques, with the exception of the raw regularized linear regression, which was prone to occasionally overfitting.

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 the 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 4.

Once again, the recurrent architectures performed best. However, 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.

4.3 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.

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, pouring at 9 degrees/second and terminating each pour at a random duration. We used 800

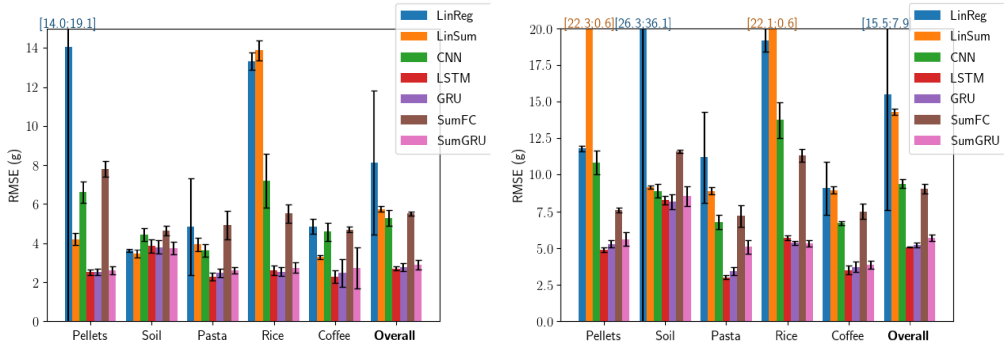

Figure 4: Performance of all-material models on estimating mass of each particular material. Bars that surpass the bounds of the graph have their inter-trial means and standard deviations respectively printed above them. **(Left)** Estimating poured mass. **(Right)** Estimating shaken mass.

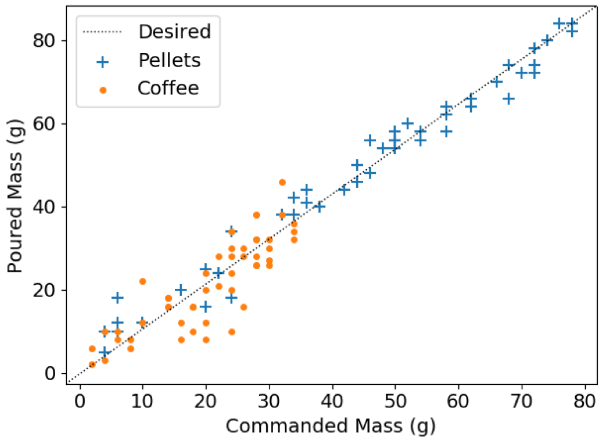

Figure 5: Controlling pours of granular material to a desired mass with a SumGRU model, using only tactile audio feedback.

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 5.

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

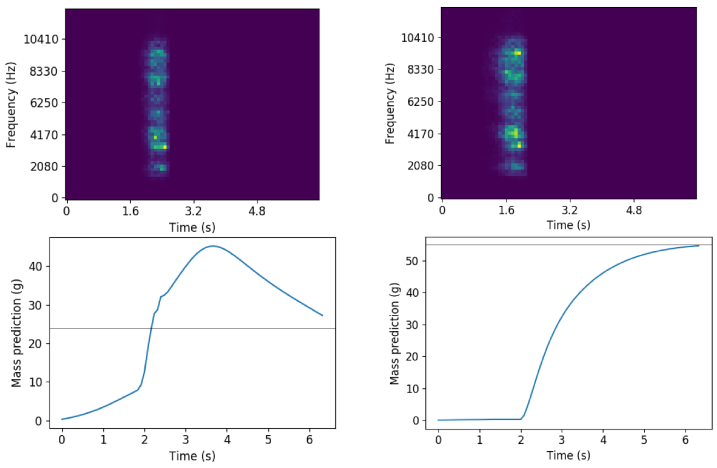

Figure 6: Demonstrating the value of the summing architectures for real-time estimation applications, when only training with weak supervision. Each plot shows the spectrogram used as input above the current estimation of mass, as output by the recurrent architecture at each time step. In the mass estimation 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.

architectures for offline estimation. However, they may be useful in real-time mass estimation applications. As shown in Figure 6, 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.

We noticed during tuning that the CNN and SumFC models were sensitive to the hyperparameter settings. 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. 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.

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.

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).

References

[1] W. H. Warren and R. 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.

[2] C. Carello, K. L. Anderson, and A. J. Kunkler-Peck. Perception of object length by sound. *Psychological science*, 9(3):211–214, 1998.

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

[4] M. 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.

[5] P. W. Cleary and M. 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.

[6] B. L. Giordano and S. 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.

[7] S. Lakatos, S. McAdams, and R. 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.

[8] T. Nakamura, T. Nagai, and N. 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.

[9] J. Sinapov, M. Wiemer, and A. 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.

[10] S. Griffith, V. Sukhoy, T. Wegter, and A. 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.

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

[12] H. P. Saal, J.-A. Ting, and S. 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.

[13] C. Schenck, J. Sinapov, D. Johnston, and A. Stoytchev. Which object fits best? solving matrix completion tasks with a humanoid robot. *IEEE Transactions on Autonomous Mental Development*, 6(3):226–240, 2014.

[14] A. Yamaguchi and C. 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.

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

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

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

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

[19] 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*, Jan. 2017.

[20] E. Sprengel, M. Jaggi, Y. Kilcher, and T. Hofmann. Audio based bird species identification using deep learning techniques. In *LifeCLEF 2016*, number EPFL-CONF-229232, pages 547–559, 2016.

[21] H. Lee, P. Pham, Y. Largman, and A. Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, pages 1096–1104, 2009.

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

[23] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[24] T. H. Vu and J.-C. Wang. Acoustic scene and event recognition using recurrent neural networks. *Detection and Classification of Acoustic Scenes and Events*, 2016, 2016.

[25] Y. Tang, Y. Huang, Z. Wu, H. Meng, M. Xu, and L. 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.

Appendix A 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 7): 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 $z_\text{i}, z_\text{f} \in [0.5, 4]$ 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 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 8 - 12.

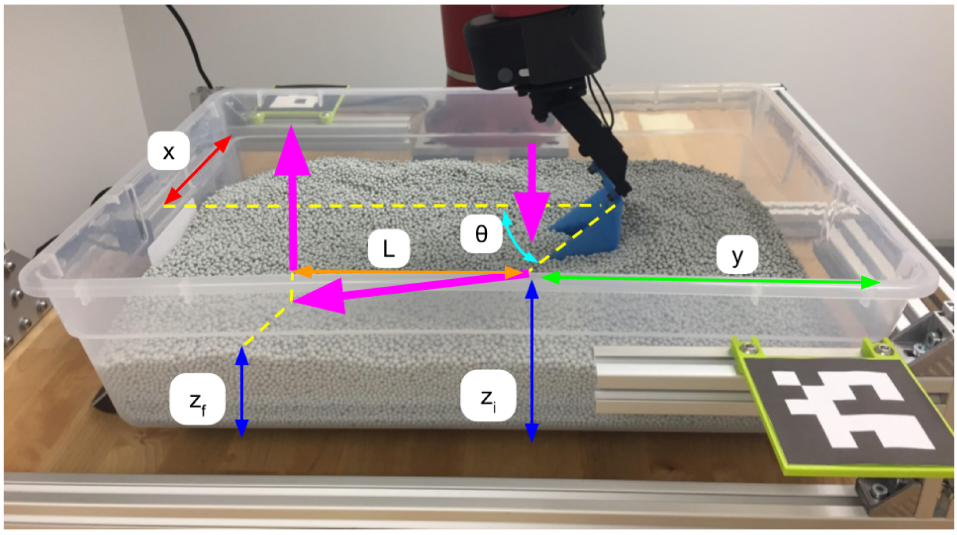

Figure 7: Parameters of scooping actions used for dataset collection.

Table 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

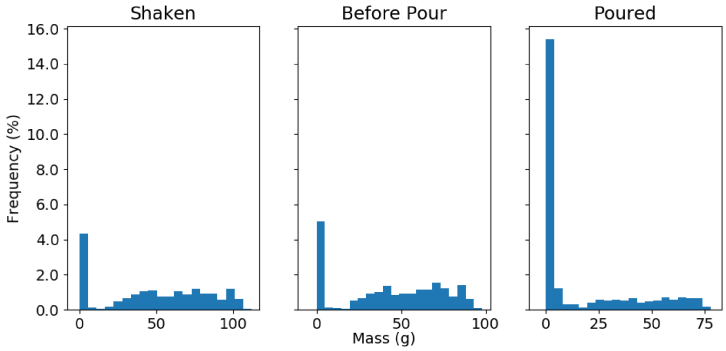

Figure 8: Distributions of masses for pellets dataset.

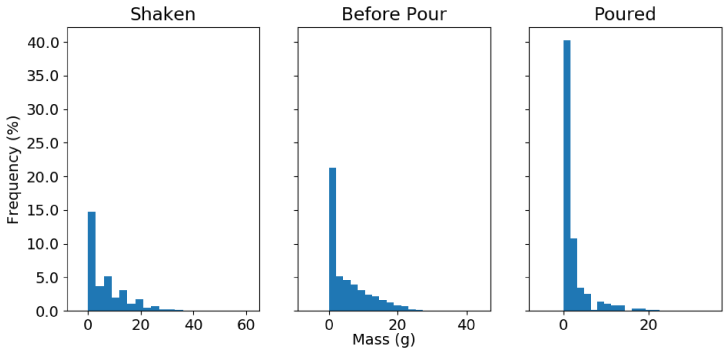

Figure 9: Distributions of masses for soil dataset.

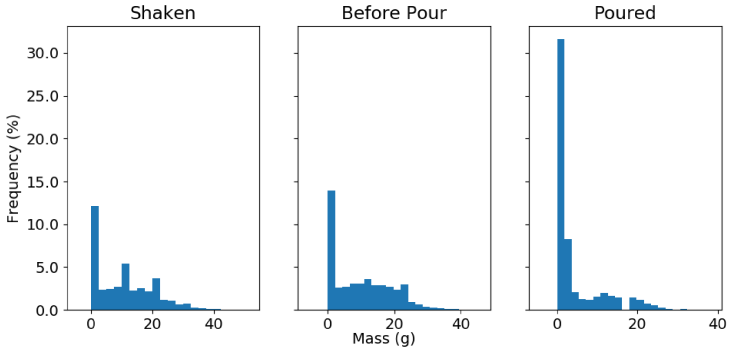

Figure 10: Distributions of masses for pasta dataset.

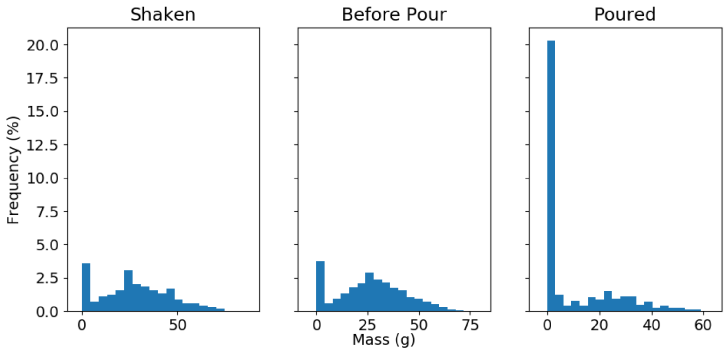

Figure 11: Distributions of masses for rice dataset.

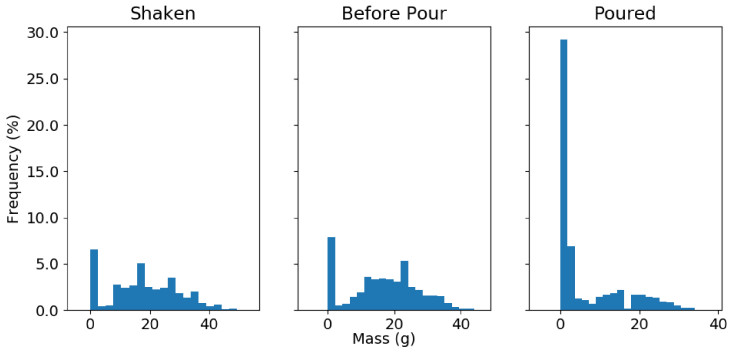

Figure 12: Distributions of masses for coffee dataset.

Appendix B Granular Material Properties

Figure 13: 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").

Images of each granular material used in our experiments are shown in Figure 13. 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 2.

Table 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

Appendix C 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 rather unaffected by ambient noise, including robot actuation noises. 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 14, 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.

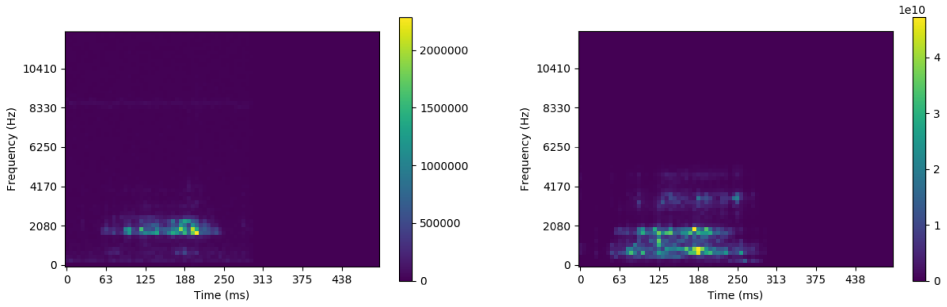

Figure 14: Spectrograms from different microphones recording simultaneously while the robot is pouring 10 g of soil. **(Left)** Spectrogram from the contact microphone used in our dataset. **(Right)** Spectrogram from a cardioid-profile lapel microphone. Note the more broad range of frequencies represented 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.

Appendix D Model Architecture Visualizations

Schematics and equations demonstrating the differences between the the LSTM and GRU memory units are shown in Figure 15. 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). Schematics of overall model architectures for all of the proposed neural network models are shown in Figure 16.

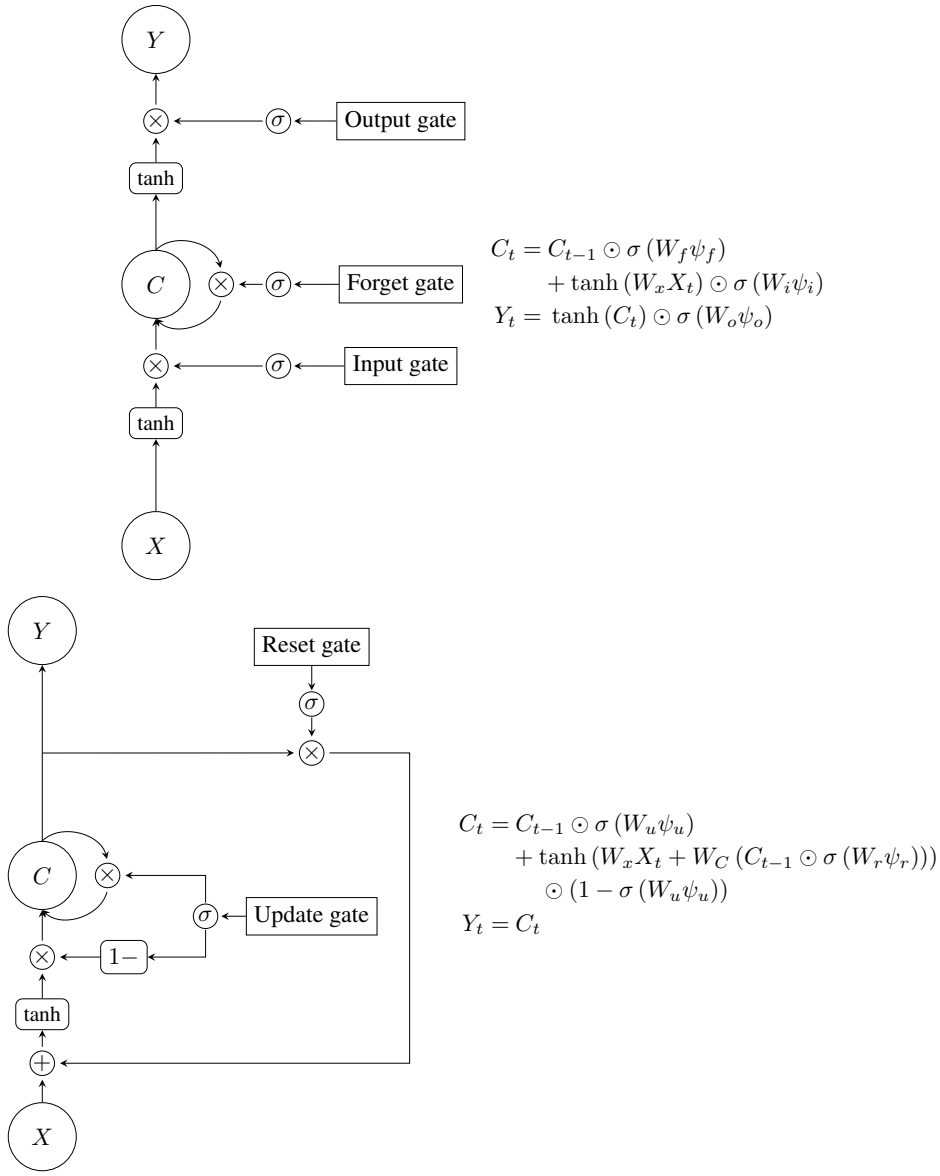

Figure 15: Comparison of (**Top**) LSTM and (**Bottom**) GRU memory cells. For both diagrams, X is the input, Y is the output, and C is the memory vector. W refers to a learned matrix. ψ refers to an input vector composed by concatenating X and Y_{t-1}. $\odot$ is the Hadamard product. σ is a sigmoid with range $[0, 1]$.

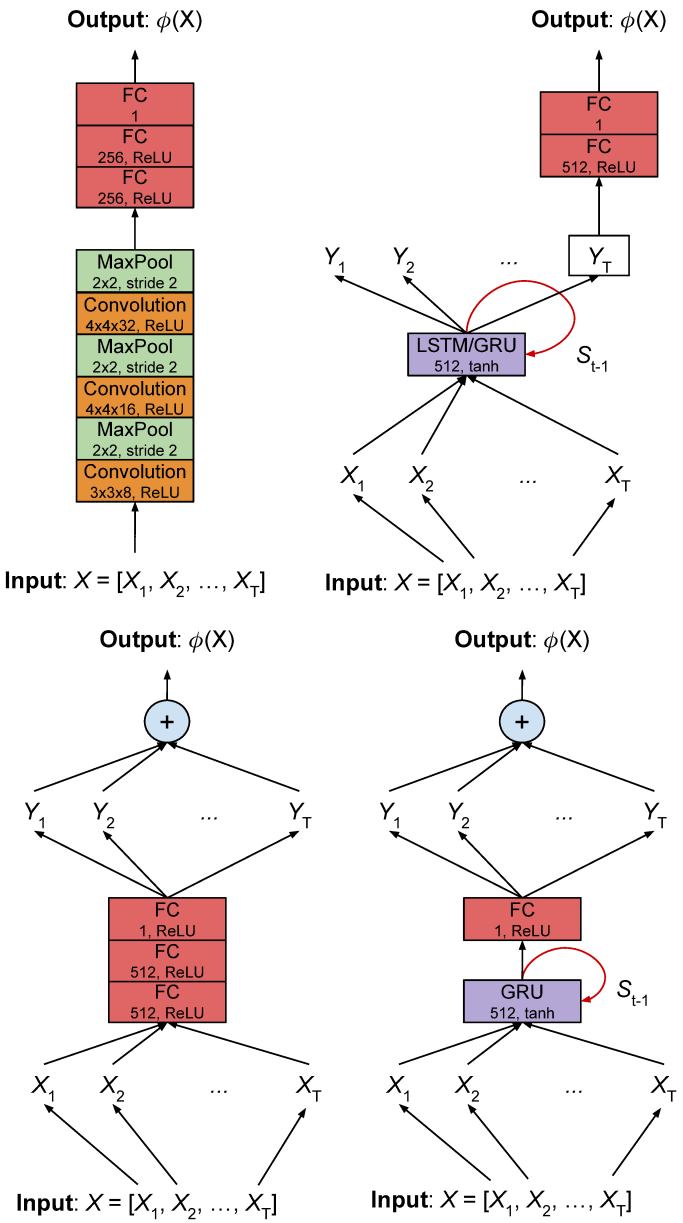

Figure 16: Schematics of model architectures. The input X is a spectrogram where the columns $\{X_1, X_2, ...X_T\}$ represent the slices of the spectrogram in its time dimension, *e.g.* X_1 is the first column vector of powers for the different frequencies in the first time bin of the spectrogram. **(Top 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).

Appendix E Generalization Evaluation

We evaluated the generalization abilities of our models in two ways. First, we tested the ability of our models to generalize to a new material for which they had not been trained. Next, we tested the ability of our models to generalize to amounts of material which were not represented in our training data.

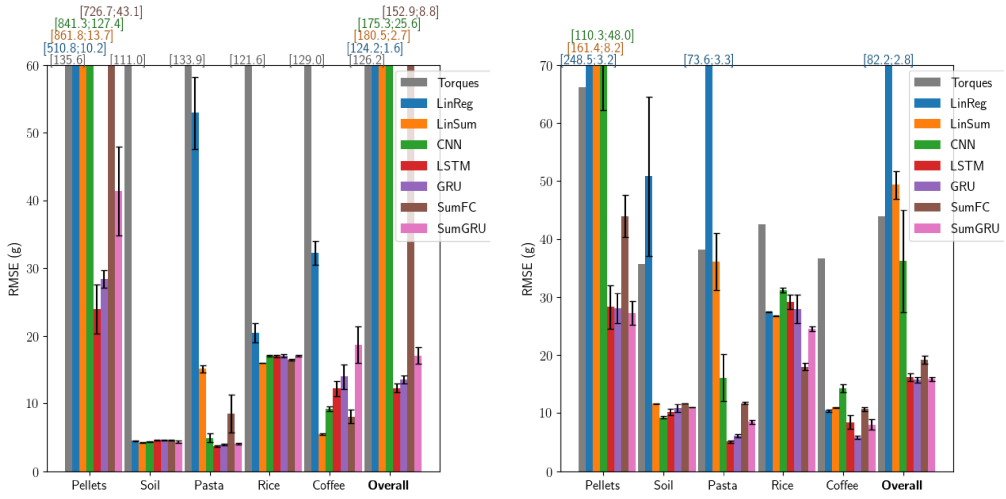

Figure 17: Performance of models on estimating mass of an untrained material, when trained on all other materials. Bars that surpass the bounds of the graph have their inter-trial means and standard deviations respectively printed above them. **(Left)** Estimating poured mass. **(Right)** Estimating shaken mass.

E.1 Generalizing to New Materials

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 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 Figure 17.

Each neural model performed surprisingly well on the soil and pasta, suggesting that the neural architectures were able to sufficiently learn features relevant to each of these materials from the data of the other materials. Overall however, 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.

E.2 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 20$^{\text{th}}$ percentile, the second fold the 20$^{\text{th}}$ to the 40$^{\text{th}}$ 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 18.

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 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.

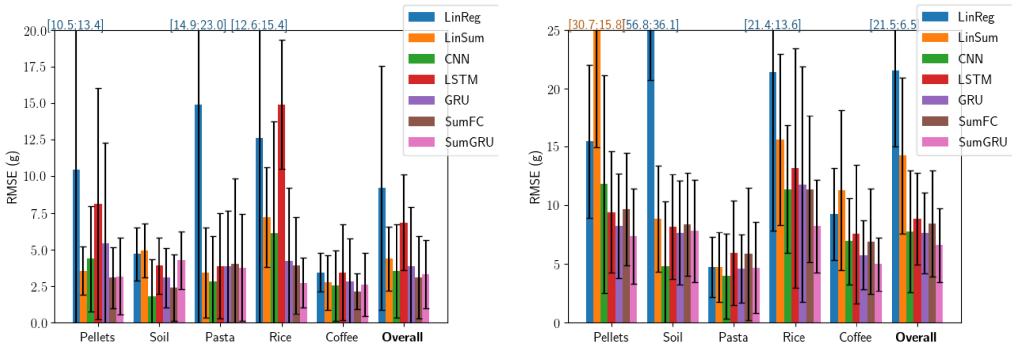

Figure 18: Performance of single-material models on estimating mass during leave-one-level-out cross-validation. Bars that surpass the bounds of the graph have their inter-trial means and standard deviations respectively printed above them. **(Left)** Estimating poured mass. **(Right)** Estimating shaken mass.

Appendix F 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 19 and 20, respectively. Individual results for each material are then broken down for both shaking and pouring and shown in Figures 21 - 30.

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

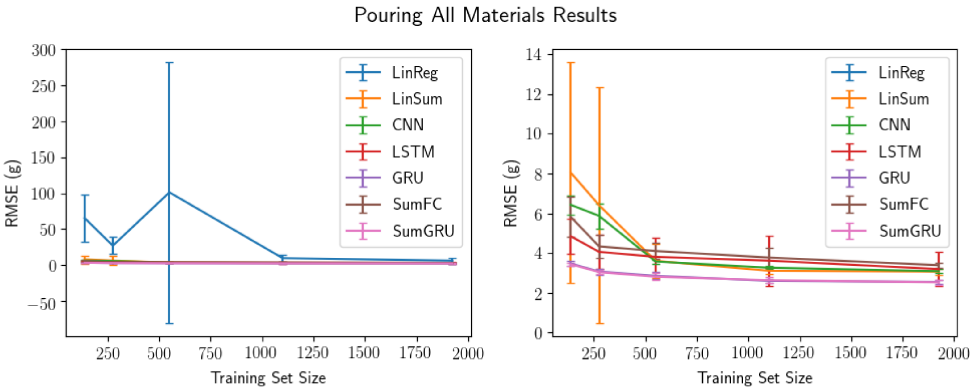

Figure 19: Training curves showing variation in performance of models with different sized training sets *averaged over data from pouring all materials*. Results are from averaging over 5 trials. **(Left)** Results from all models. **(Right)** Results from regularized linear regression omitted to show detail of other models' performances.

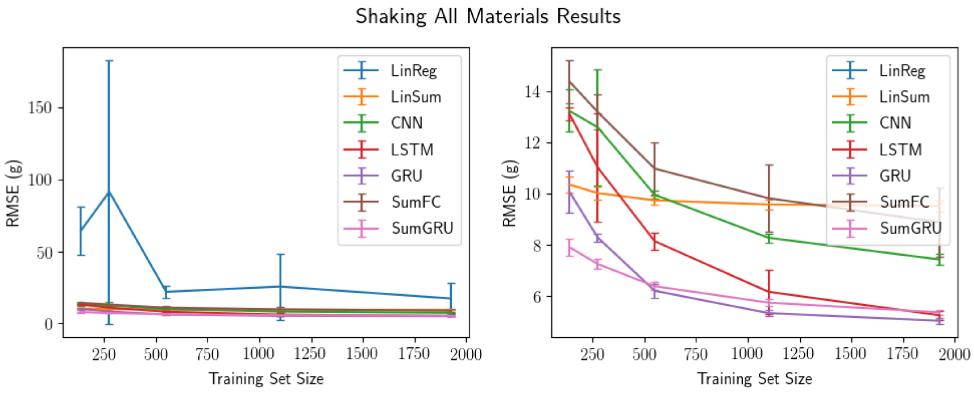

Figure 20: Training curves showing variation in performance of models with different sized training sets *averaged over data from shaking all materials*. Results are from averaging over 5 trials. **(Left)** Results from all models. **(Right)** Results from regularized linear regression omitted to show detail of other models' performances.

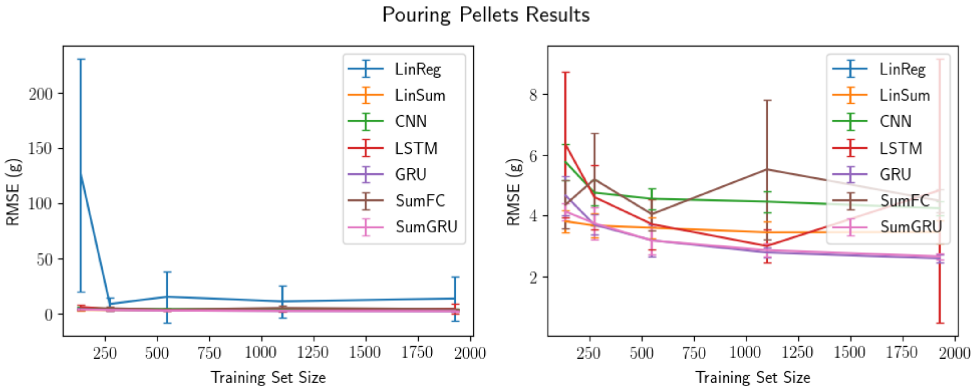

Figure 21: Training curves showing variation in performance of models with different sized training sets of *pouring pellets data*. Results are from averaging over 5 trials. **(Left)** Results from all models. **(Right)** Results from regularized linear regression omitted to show detail of other models' performances.

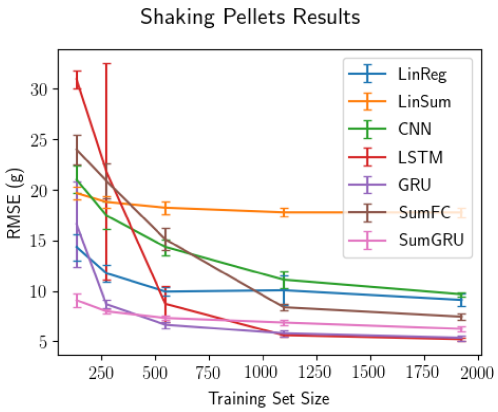

Figure 22: Training curves showing variation in performance of models with different sized training sets of *shaking pellets data*. Results are from averaging over 5 trials.

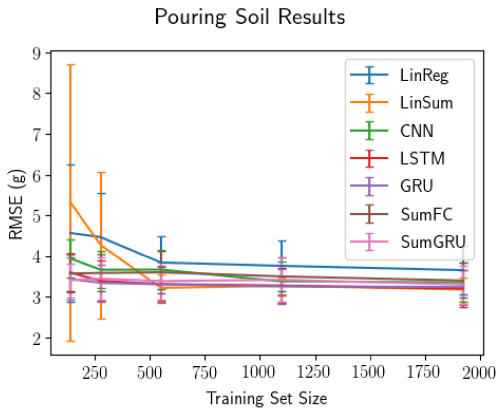

Figure 23: Training curves showing variation in performance of models with different sized training sets of *pouring soil data*. Results are from averaging over 5 trials.

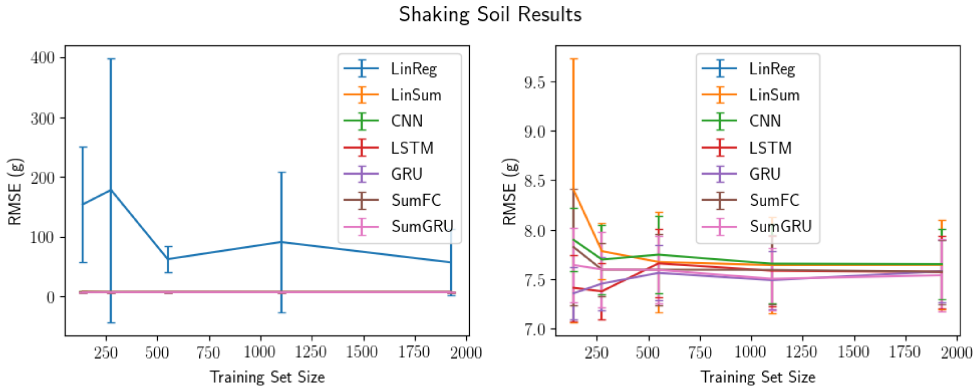

Figure 24: Training curves showing variation in performance of models with different sized training sets of *shaking soil data*. Results are from averaging over 5 trials. **(Left)** Results from all models. **(Right)** Results from regularized linear regression omitted to show detail of other models' performances.

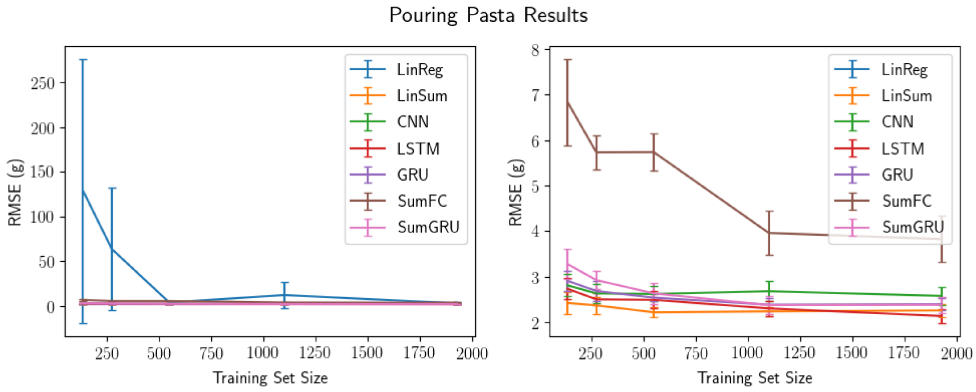

Figure 25: Training curves showing variation in performance of models with different sized training sets of *pouring pasta data*. Results are from averaging over 5 trials. **(Left)** Results from all models. **(Right)** Results from regularized linear regression omitted to show detail of other models' performances.

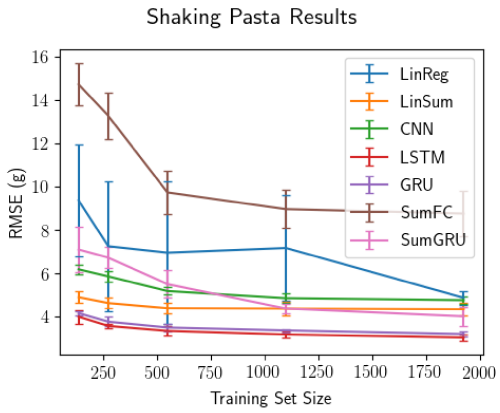

Figure 26: Training curves showing variation in performance of models with different sized training sets of *shaking pasta data*. Results are from averaging over 5 trials.

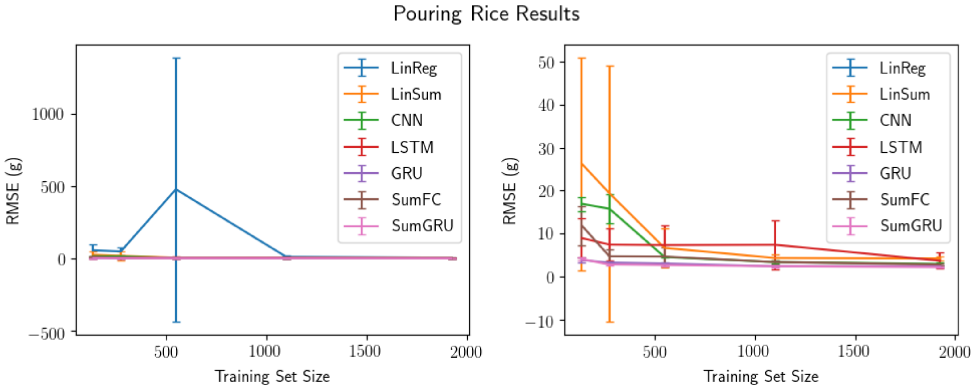

Figure 27: Training curves showing variation in performance of models with different sized training sets of *pouring rice data*. Results are from averaging over 5 trials. **(Left)** Results from all models. **(Right)** Results from regularized linear regression omitted to show detail of other models' performances.

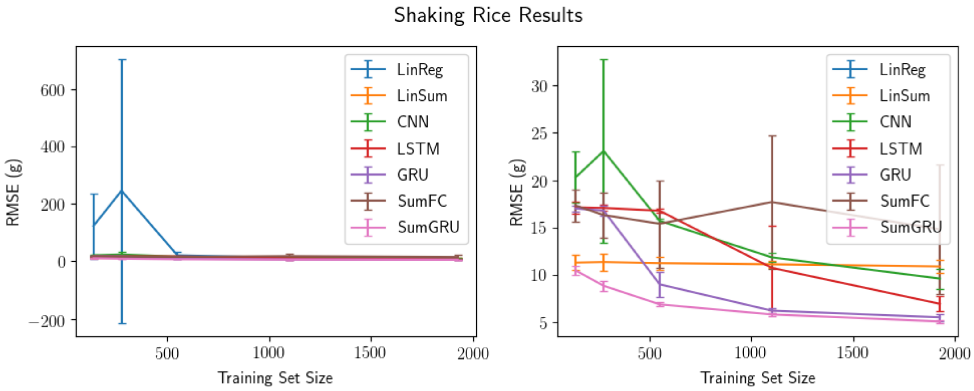

Figure 28: Training curves showing variation in performance of models with different sized training sets of *shaking rice data*. Results are from averaging over 5 trials. **(Left)** Results from all models. **(Right)** Results from regularized linear regression omitted to show detail of other models' performances.

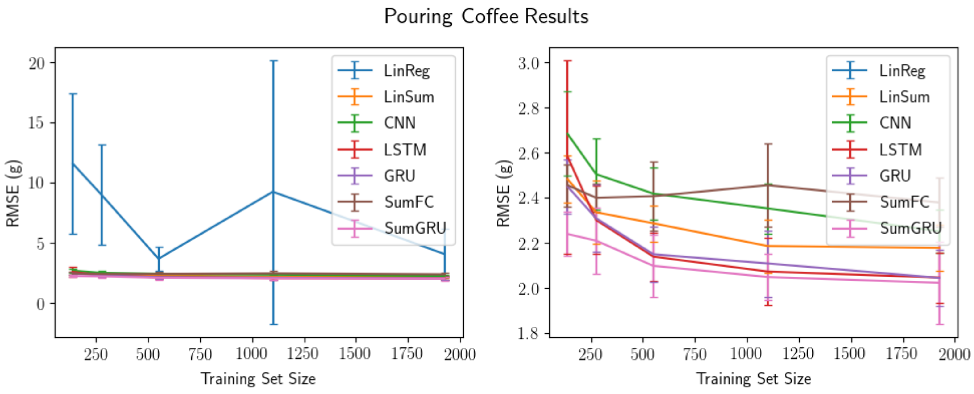

Figure 29: Training curves showing variation in performance of models with different sized training sets of *pouring coffee data*. Results are from averaging over 5 trials. **(Left)** Results from all models. **(Right)** Results from regularized linear regression omitted to show detail of other models' performances.

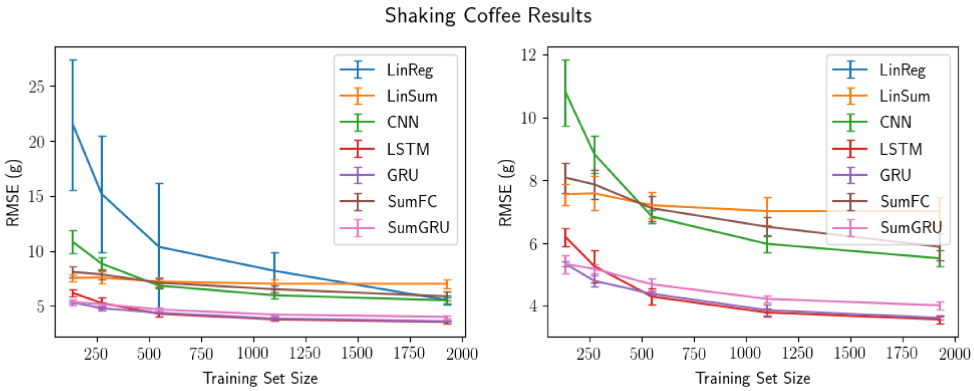

Figure 30: Training curves showing variation in performance of models with different sized training sets of *shaking coffee data*. Results are from averaging over 5 trials. **(Left)** Results from all models. **(Right)** Results from regularized linear regression omitted to show detail of other models' performances.