Hearing the Slide: Acoustic-Guided Constraint Learning for Fast Non-Prehensile Transport

Yuemin Mao[1], Bardienus P. Duisterhof[1], Moonyoung Lee[1], Jeffrey Ichnowski[1]

***Abstract*— Object transport tasks are fundamental in robotic automation, emphasizing the importance of efficient and secure methods for moving objects. Non-prehensile transport can significantly improve transport efficiency, as it enables handling multiple objects simultaneously and accommodating objects unsuitable for parallel-jaw or suction grasps. Existing approaches incorporate constraints based on the Coulomb friction model, which is imprecise during fast motions where inherent mechanical vibrations occur. Imprecise constraints can cause transported objects to slide or even fall off the tray. To address this limitation, we propose a novel method to learn a friction model using acoustic sensing that maps a tray's motion profile to a dynamically conditioned friction coefficient. This learned model enables an optimization-based motion planner to adjust the friction constraint at each control step according to the planned motion at that step. In experiments, we generate time-optimized trajectories for a UR5e robot to transport various objects with constraints using both the standard Coulomb friction model and the learned friction model. Results suggest that the learned friction model reduces object displacement by up to 86.0 % compared to the baseline, highlighting the effectiveness of acoustic sensing in learning real-world friction constraints. Video demonstrations see https://fast-non-prehensile.github.io/.**

I. Introduction

When using a tray as the end effector, robots avoid issues around parallel jaw grippers where objects can be too large for the gripper [1], difficult-to-grasp [2], or irregularly-shaped [3]. Similarly, trays avoid issues around suction grasping porous or deformable objects, or computing suction-grasp stability [4], [5]. Additionally, non-prehensile transport can boost efficiency by allowing simultaneous transport of multiple objects [6], [7], reduce computational costs by avoiding computations for traditional rigid grasps, and lower upfront and maintenance expenses as trays have no actuation.

We consider a modular workcell setup where a robotic manipulator uses a tray to transport objects, adaptable to a wide range of automated manufacturing scenarios that benefit from fast object handling. A key safety concern, especially when moving large or fragile objects, is minimizing any displacement of the object on the tray, which requires balancing inertial and frictional forces on the transported objects. Recent works have addressed the challenges of fast non-prehensile transport by incorporating constraints based on the Coulomb friction model with model predictive control (MPC) to secure objects [8], [9]. However, they do not explicitly aim to push the robots to their speed limit or minimize object sliding, which occurs when the Coulomb friction model fails to account for some real-world factors during fast transport. We address this limitation by augmenting the Coulomb friction model. We analyze the dynamic interactions between the tray and transported objects to develop a friction model that infers the real-world friction constraint, allowing a robot to simultaneously minimize object sliding and transport time.

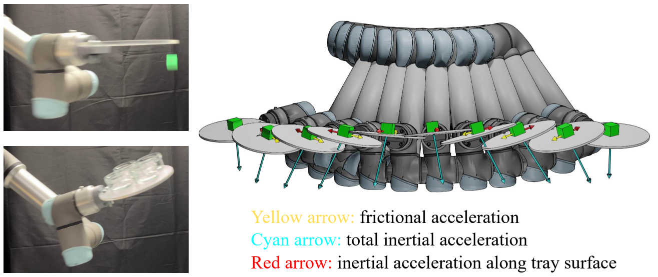
Yellow arrow: frictional acceleration
Cyan arrow: total inertial acceleration
Red arrow: inertial acceleration along tray surface

Fig. 1: **Fast non-prehensile transport**. **Top left:** Time-optimized transport without constraints drops the object. **Bottom left:** With friction-based constraints, the motion planner initiates tray tilt to compensate for inertia effects, safely transporting multiple fragile objects. **Right:** We propose a learned dynamic constraint to balance the inertial and frictional forces during fast transport, reducing object sliding.

Learning a dynamic friction model requires observing transported objects throughout high-speed motions. While camera-based setups are common in workcells, they require special hardware to have a high frame rate, avoid motion blur, and work around occlusions. While augmenting the tray with pressure-based tactile sensors would provide high-speed sensing that avoids occlusions, it comes at a significant additional expense and the potential for wear over time. Instead, we propose using acoustic sensing as it offers advantages including high sampling frequency, low latency, robustness to environmental noise, and ease of implementation, thus well-suited for detecting object sliding in non-prehensile transport. By attaching a contact microphone to the tray, we capture vibrations induced by object motion everywhere on the tray surface.

During high-speed motions constrained by the *standard* Coulomb friction model, the attached contact microphone detects unexpected sliding motions correlated with the robot's mechanical vibrations, which intensify with transport speed. Leveraging the sensing data, we train a neural network to learn a friction coefficient that updates with varying tray velocity and acceleration, improving non-prehensile transport safety beyond limitations of methods based on the standard Coulomb friction model.

In experiments, we learn the dynamic friction model with

[1] Robotics Institute, Carnegie Mellon University. {yueminm, jeffi}@andrew.cmu.edu

data collected from a UR5e robot. We test constraints with the standard Coulomb friction model and the learned friction model on 8 objects with 12 different configurations. Results suggest that the learned friction model can reduce the object displacement by up to 86.0 % compared to the baseline.

The contributions of this paper are:

- An acoustic sensing approach to learn effective friction between the tray and transported objects during high-speed non-prehensile transport.
- An integration of the learned friction model into a time-optimizing motion planner.
- Physical experiments on a UR5e robot suggesting the learned friction model lowers displacement over a standard Coulomb friction model during high-speed non-prehensile transport.

II. Related Work

A. Non-Prehensile Transport

The Waiter's Problem, proposed by Bernheisel and Lynch [10], is the problem formulation most closely related to this work. It proposes transport of objects subject to unilateral constraints only and motion planning that takes into account inertial effects. Different approaches have been taken to secure objects on a moving tray [11]. Acharya et al. [12] focus on preventing object tipping and propose using s-curve motion profiles to control peak accelerations after analyzing the marginal stability of objects. Zhou et al. [13] transports a tray with a dual-arm setup and employ MPC to compute grasping wrench, together with a force servoing controller to control contact forces between the tray and both end effectors. Selvaggio et al. [8] design an MPC framework to move the tray along a reference trajectory. Heins and Schoellig [9] develop an MPC algorithm for a mobile manipulator to respond to dynamic obstacles while balancing the objects by optimizing trajectories online.

Unlike the previous works that primarily address the motion planning challenges of non-prehensile transport, our approach focuses on the performance of transport from the aspect of object displacement relative to the tray during transport. We push friction-based constraints to the real-world limits by addressing physical phenomena rather than relying solely on standard physics analysis of Coulomb friction. Instead of using minimum statically-feasible friction coefficients [9] or a conservatively approximated friction cone space which may still lead to significant ($\geq$ 1 cm) object displacements during transport [8], we measure the friction coefficients experimentally, use their exact values during motion planning for time optimization, and refine them with a learned model to boost performance.

B. Acoustic Sensing in Manipulation

Acoustic sensing with contact microphones provides a low-cost solution for ensuring that signals are accurately aligned in time with little engineering effort, making data processing easier. In manipulation tasks, contact microphone data can facilitate perception and learning [14], [15], [16]. Clarke et al. [17] propose using vibrations from contact to measure flow and amount of granular materials during scooping and pouring using a signal processing method that converts audio into binned spectrograms. We adapt their approach to focus on detecting sudden increases in vibration intensity caused by object sliding, rather than training a model directly from the spectrograms' visual features.

C. Motion Planning for Dynamic Manipulation

Optimization-based motion planners generate smooth trajectories while enforcing constraints [18]–[20]. In the context of dynamic manipulation, the kinematic and dynamic limitations of robots play an important role in the formulation of constraints to speed up motions [21], [22]. Additionally, the constraints should incorporate either analytical formulations of the problems [23]–[27], or learned models [5], [28]–[31], to account for the dynamic properties during motion planning. Inspired by these prior works, we take an integrated approach for motion planning. We adapt the optimization-based motion planning framework from GOMP-FIT [26], formulate a task-oriented constraint analytically, and embed a learned model in this constraint.

III. Problem Statement

A robot with n degrees of freedom has a tray end effector. Let $\mathbf{q} \in \mathcal{C}$ be the angular configurations of all n joints of the robot, where $\mathcal{C} \subset \mathbb{R}^n$ is the set of all valid configurations. Let $\mathbf{x}_i = [\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i] \in \mathcal{X}$ be the robot's dynamic state at any time step i. Let $\tau = (\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_H)$ be the complete trajectory discretized to $H + 1$ time steps. Let $f_k : \mathcal{C} \rightarrow SE(3)$ be the forward kinematics function that computes the pose of the tray.

Given the start and goal tray positions, $\mathbf{g}_{\text{start}}, \mathbf{g}_{\text{goal}} \in SE(3)$, and the centroids of m objects resting on the robot's tray, the objective is two-fold. First, identify task-oriented dynamic constraints $\mathcal{D}$ adaptable to real-world uncertainties and a function f_c to map robot's state to the constrained parameters for each object to prevent sliding. Second, compute a time-optimized trajectory τ that satisfies the following conditions: $f_k(\mathbf{q}_0) = \mathbf{g}_{\text{start}}$, $f_k(\mathbf{q}_H) = \mathbf{g}_{\text{goal}}$, while ensuring that the robot kinematic and dynamic constraints, along with the identified constraints $\mathcal{D}$, are met at each time step.

For objects transported on a moving tray, we assume: 1) they have even contact surfaces, 2) they will not topple during transport, 3) all external forces are exerted at their centroids, and 4) they do not apply forces on each other. Additionally, we assume aerodynamic, adhesive, and electrostatic forces to be negligible during object transport.

IV. Preliminary

We construct the motion planning framework for evaluating various friction models in fast non-prehensile transport. We formulate a baseline constraint with the standard Coulomb friction model and integrate it with the motion planner from GOMP-FIT [26] to solve the non-prehensile transport problem. This section provides background on GOMP-FIT, then discusses the integration of task-specific constraints.

A. GOMP-FIT Background

GOMP-FIT formulates motion planning for fast transport as an optimization problem and solves it with a sequential quadratic program (SQP) solver. GOMP-FIT discretizes the trajectory into a sequence of $H+1$ waypoints consisting of joint configurations and their derivatives, $\mathbf{x}_i = [\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i], i \in [0,\ H]$, each separated by a timestep t_{step}.

We employ the inner loop of GOMP-FIT that solves an SQP where the optimization is in the form:

$$\begin{aligned}
\arg\min_{\mathbf{x}_{[0\ldots H]}} \quad & \frac{1}{2}\mathbf{x}^T P \mathbf{x} + \mathbf{p}^T \mathbf{x} & \\
\text{s.t.} \quad & f_k(\mathbf{q}_0) = \mathbf{g}_{\text{start}}, f_k(\mathbf{q}_H) = \mathbf{g}_{\text{goal}} & \\
& \mathbf{q}_i \in \mathcal{C} & \forall i \in [0,\ H] \\
& \mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i, \dddot{\mathbf{q}}_i \in \text{joint limits} & \forall i \in [0,\ H] \\
& \mathbf{q}_{i+1} = \mathbf{q}_i + t_{\text{step}}\dot{\mathbf{q}}_i + \frac{1}{2}t^2\ddot{\mathbf{q}}_i & \forall i \in [0,\ H) \\
& \dot{\mathbf{q}}_{i+1} = \dot{\mathbf{q}}_i + t_{\text{step}}\ddot{\mathbf{q}}_i & \forall i \in [0,\ H) \\
& \dddot{\mathbf{q}}_{i+1} = \left(\ddot{\mathbf{q}}_{i+1} - \ddot{\mathbf{q}}_i\right)/t_{\text{step}} & \forall i \in [0,\ H) \\
& f_c(\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i) \in \mathcal{D} & \forall i \in [0,\ H]\,,
\end{aligned}$$

where P minimizes the sum-of-squared jerks, $\mathbf{p}$ includes penalties for non-convex constraints, and $\mathcal{D}$ and f_c together characterize the task-oriented constraints in dynamic manipulation. The SQP solver repeatedly linearizes the non-linear constraints to form a quadratic program (QP) and accepts the solutions that improve the trajectory. In the $(k+1)$ iterate, a non-linear constraint of the form $g(\mathbf{x}) \leq c$ is linearized around the current iterate $\mathbf{x}^k$ through first-order approximation using its Jacobian: $J\mathbf{x}^k{+}1 \leq J\mathbf{x}^k{-}g(\mathbf{x}^k){+}c$.

We apply a time shrinking strategy similar to Curobo [32], keeping a fixed H (H=32 in experiments) and shrinking t_{step} through binary search (starting from $t_{\text{step}} = 0.25\,\text{s}$ in experiments). The algorithm terminates when the SQP fails to find a solution or when the update of t_{step} between iterations is less than 50 µs. The planner returns the trajectory with the smallest t_{step} that allows the SQP to succeed as (locally) time-optimal.

B. Baseline Constraint with Standard Coulomb Friction Model

The formulation of the baseline constraint is based on the friction and inertia at the centroid of each object. We derive the maximum static frictional force from the Coulomb Friction Model: $F_f \leq \mu_s F_n$, where F_f and F_n are the frictional force and normal force exerted by each contact surface on the other respectively, and μ_s is the static friction coefficient measured experimentally. We compute the acceleration due to gravitational, Euler, Coriolis, and centrifugal forces at each object's centroid, using the Recursive Newton Euler (RNE) method [33].

When an object maintains complete contact with the tray, it remains stationary relative to the tray if the frictional force from the tray balances the inertial force tangential to the tray surface. Otherwise, it slides. We thus constrain trajectories to keep the inertial force along the tray surface below the maximum static frictional force. The constraint has the form:

$$\begin{aligned}
& m\left\|\mu_s\left(\mathbf{a}_i \cdot f_n(\mathbf{q}_i)\right)\right\| \\
& - m\left\|\mathbf{a}_i - \left(\mathbf{a}_i \cdot f_n(\mathbf{q}_i)\right) f_n(\mathbf{q}_i)\right\| \geq 0, \quad \forall i \in [0,\ H]\,,
\end{aligned} \tag{1}$$

where m is the object mass, $\mathbf{a}_i$ is the inertial acceleration at time step i, and $f_n(\mathbf{q}_i)$ is the normal of the tray's contact surface at robot configuration $\mathbf{q}_i$.

One-Shot Measurement of Static Friction Coefficient: To derive the static friction coefficient, we assume only gravitational, normal, and frictional forces act on the object when the tray is stationary. An object begins to slide on the tray inclined at angle θ when the total force along the tray surface reaches equilibrium: $\mu_s(mg\cos\theta) - mg\sin\theta = 0 \Rightarrow \mu_s = \tan\theta$, where m is the object mass and g is gravity. Thus, we derive μ_s from the minimum θ that induces object sliding. To measure $\theta_{\min}$, we increment θ by 0.01 radians until the object starts to slide. In experiments, we measure μ_s for each material, and each measurement takes less than 1 minute.

V. Method

Real-robot experiments with the baseline constraint show significant object displacement, indicating additional factors not characterized by the Coulomb friction model. Thus, we investigate the causes of object sliding during fast nonprehensile transport and propose a learning-based method to reduce object displacement.

A. Motion Characterization with Acoustic Sensing

We employ acoustic sensing with a passive piezoelectric microphone, known as a contact microphone, as it offers a direct and robust approach to detecting object displacement on a fast moving tray. When an object slides on the tray, it generates mechanical vibrations that propagate through the material. A contact microphone captures these vibrations from any point on the moving tray at a sampling frequency of 44.1 kHz, creating a high-fidelity mapping between the physical event and the sensor output.

Experiments with a contact microphone attached to an empty tray show that tray vibrations intensify with the increase of the robot end-effector velocity (Fig. 2), even with low-jerk motions. These vibrations cause rapid fluctuations in the normal and tangential forces at the contact surfaces, resulting in uncertainty in the frictional force that holds objects in place. This suggests that factors contributing to object sliding that are not characterized by the standard Coulomb friction model become more significant as the tray speeds up. Thus, we propose learning a motion-aware friction model.

B. Constraint with Learned Friction Model

We use $\tilde{\mu}$, a dynamically conditioned friction coefficient, to take into account external forces not characterized by the static friction coefficient μ_s, minimizing object sliding. The updated friction model has the form: $F_{t_{\max}} = \tilde{\mu} F_n$, where F_n is the normal force and $F_{t_{\max}}$ is the maximum tangential

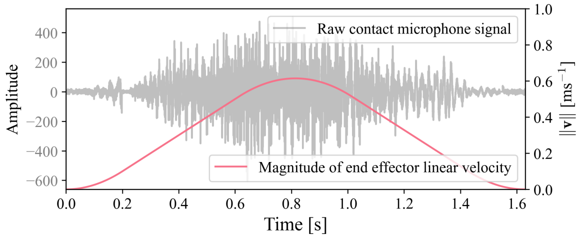

Fig. 2: Raw contact-microphone signal collected when a UR5e robot's tray end effector accelerates at a maximum jerk of $\pm 4\,\mathrm{ms}^{-3}$, with no object on it. The overlay of acoustic signal amplitude and end-effector linear speed shows a positive correlation between the tray's vibration and speed.

force, both exerted on the object's contact surfaces by the tray. Here, $\tilde{\mu}$ changes with the robot's end-effector motion, but, at any moment in time, $\tilde{\mu}$ is constant anywhere on the tray.

As tray vibration increases with velocity, and is highly correlated to sliding (Sec. V-A), we train a dynamic parameter α for a specific robot to accommodate the vibrations inherent in its mechanical design such that $\tilde{\mu} = \alpha\mu_s$. The constraint from Eq. 1 now has the form:

$$\begin{aligned}&\|\alpha_i\mu_s\left(\mathbf{a}_i \cdot f_n\left(\mathbf{q}_i\right)\right)\| \\ &- \|\mathbf{a}_i - \left(\mathbf{a}_i \cdot f_n(\mathbf{q}_i)\right) f_n(\mathbf{q}_i)\| \geq 0, \quad \forall i \in [0,\, H]\,.\end{aligned} \tag{2}$$

C. Model Training with Acoustic Sensing

We apply a trial-and-error strategy during data collection to identify the threshold features when the object starts sliding. Similar to the GOMP-ST data-collection pipeline [5], we use Ruckig [34] to plan a series of time-optimal straight-line horizontal motions for the tray between 2 fixed points, with varying acceleration and jerk limits. The robot keeps the tray level throughout the motion, with both linear velocity and acceleration aligned with the direction of movement. This alignment ensures consistent linear velocity and acceleration across the tray, eliminating the need to reposition the object after each motion. We then translate the motions into joint-space trajectories with an analytic inverse kinematics solver. We execute each trajectory 5 times without the object and 5 times with the object on the tray, recording contact microphone signals.

For data analysis, we use the Python audio package Librosa [35] and SciPy [36] to process raw signals and the Noisereduce Python package [37] to extract signals representing the dynamic interactions between the object and the tray. In this process, we treat signals from trials with the object as the sound, and signals from trials without the object as the non-stationary noise [38]. We then process the filtered signals into a binned spectrogram using the method from Clarke et al. [17], with a time bin size matching the robot controller's time step and a frequency bin size of 100 Hz. We identify the time t_{sliding} when the signal first shows a significantly greater magnitude and correlate it with the velocity $\mathbf{v}_{\text{sliding}}$ and acceleration $\mathbf{a}_{\text{sliding}}$ in the planned Cartesian-space trajectory at that specific time step (Fig. 3). These pairs of $\mathbf{v}_{\text{sliding}}$ and $\mathbf{a}_{\text{sliding}}$ are identified as threshold features of object sliding.

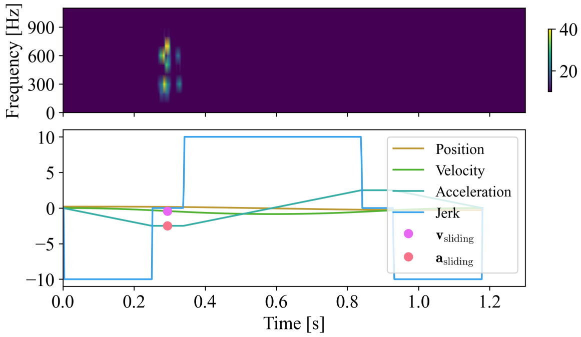

Fig. 3: Synchronization of a fully preprocessed spectrogram with the planned trajectory in Cartesian space. We focus on the first occurrence of sliding, and thus mask out the part of spectrogram past the maximum velocity. The plot shows a trial with a 0.5 m straight-line horizontal end-effector trajectory with an acceleration limit of 2.5 ms^{-2} and a jerk limit of 10 ms^{-3}. Here $t_{\text{sliding}} = 0.294$ s, $\|\mathbf{a}_{\text{sliding}}\| = 2.5$ ms^{-2}, $\|\mathbf{v}_{\text{sliding}}\| = 0.423$ ms^{-1}.

After data analysis, we further process $\mathbf{a}_{\text{sliding}}$ into α_{training} with the assumption that horizontal inertial force and gravitational force are the only forces exerted on the object during the straight-line horizontal motions:

$$m\,\|\mathbf{a}_{\text{sliding}}\| = m\alpha_{\text{training}}\mu_s\,\|\mathbf{g}\| \;\Rightarrow\; \alpha_{\text{training}} = \frac{\|\mathbf{a}_{\text{sliding}}\|}{\mu_s\,\|\mathbf{g}\|},$$

where $\mathbf{g}$ is the gravity, m is the object mass, and μ_s is the measured static friction coefficient. We then train a function $\alpha = g(\|\mathbf{v}\|)$ that computes the motion-aware friction coefficient, where $\mathbf{v}$ is the tray's linear velocity. Therefore, we formulate the complete constraint from Eq. 2 as:

$$\begin{aligned}&\|\mu_s g(f_{\|v\|}(\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i))\left[f_a(\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i) \cdot f_n(\mathbf{q}_i)\right]\| \\ &- \|f_a(\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i) - \left[f_a(\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i) \cdot f_n(\mathbf{q}_i)\right] f_n(\mathbf{q}_i)\| \geq 0, \\ &\forall i \in [0, H],\end{aligned}$$

where μ_s is the measured static friction coefficient from Sec. IV-B, $g : \mathbb{R} \to \mathbb{R}$ is the learned function that computes the dynamic constraint parameter α from the magnitude of linear velocity, $f_{\|v\|} : \mathcal{X} \to \mathbb{R}$ computes the magnitude of linear velocity, and $f_a : \mathcal{X} \to \mathbb{R}^3$ computes the linear inertial acceleration with RNE method [33]. $f_n : \mathbb{R}^n \to \mathbb{R}^3$ computes the normal vector of the tray attached to a robot arm with n degrees of freedom, which is the same anywhere on the tray. In implementation, we vectorize the constraint to compute for multiple transported objects in parallel.

VI. Experiments

We conduct real-world experiments on a UR5e robotic manipulator with a custom tray as its end effector. We measure μ_s for each tested material, collect data and train dynamic friction models using 2 different objects, and then generate trajectories to transport objects.

A. Model Training

During data collection, we use a round acrylic sheet with a 3D-printed PLA boundary as the tray (Fig. 4a). This setup allows for automatic data collection without manual resets as the boundary prevents objects from falling. We attach a

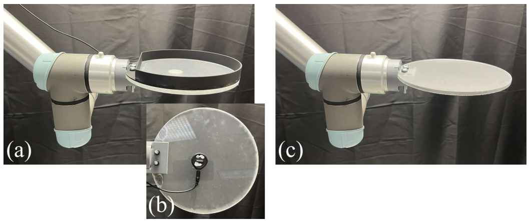

Fig. 4: (a) Data collection setup. (b) Contact microphone attached under the tray. (c) Model evaluation setup. We leave the protective film on the tray to prevent the electrostatic friction caused by acrylic.

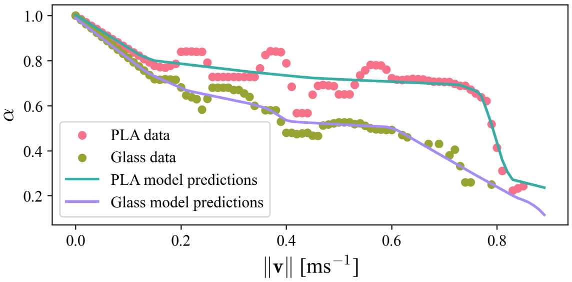

Fig. 5: Visualization of the models trained for dynamic constraint parameter α.

AMUMU SBT-10 Trans-HD Transducer at the center of the tray's bottom surface (Fig. 4b) and use Lavalier USB UL20 to amplify the signals and convert them to USB signals.

We conduct data collection with a 3D-printed PLA cube (Fig. 6a) and a small glass container (Fig. 6c). We generate straight-line horizontal trajectories between the start position $(0.20, 0.60, 0.20)$ and the end position $(-0.30, 0.60, 0.20)$ in meters relative to the robot base frame, using Ruckig with acceleration limits in [1.0, 2.5] ms^{-2} and jerk limits in [1.0, 10.0] ms^{-3}. We identify 85 unique pairs of $\|\mathbf{v}_{\text{sliding}}\|$ and $\|\mathbf{a}_{\text{sliding}}\|$ from the PLA cube and 70 unique pairs from the glass container. We augment the data by assuming $\|\mathbf{a}_{\text{sliding}}\| = \mu_s\|\mathbf{g}\|$ when $\|\mathbf{v}_{\text{sliding}}\| = 0$ and perform linear interpolation to the minimum identified $\|\mathbf{v}_{\text{sliding}}\|$. We then train an MLP model for α using the data from each object, consisting of 3 fully connected layers with LeakyReLU activations and dropouts for regularization. Fig. 5 shows the models that reduce α as velocity increases to compensate for the impact of forces under stronger vibration, with the glass function being more conservative. We test each model separately to evaluate the performance.

B. Model Evaluation

We use an acrylic sheet without boundary (Fig. 4c) during constraint evaluation experiments, so the objects will fall if the constraint fails to keep them in place. We mount a top-down camera above the tray's target position to capture objects' final positions after each motion. We attach a sticker with 5×5 mm^2 grid patterns at the edge of the tray's top surface as a reference for real-word distance when computing object displacements using a computer vision algorithm. We select 8 objects (Fig. 6) and conduct 12 sets of experiments by varying object configurations (single vs. multiple) and contact surfaces (hard vs. soft for Fig. 4h). Particularly, we choose objects with maximum grasp difficulty from the evaluation set of EGAD! [3] (Fig. 6d) to demonstrate the tray's capability of handling irregular-shaped objects. And we 3D print them with PLA to test whether complex geometries affect the constraint performance when comparing with the PLA cube (Fig. 6a).

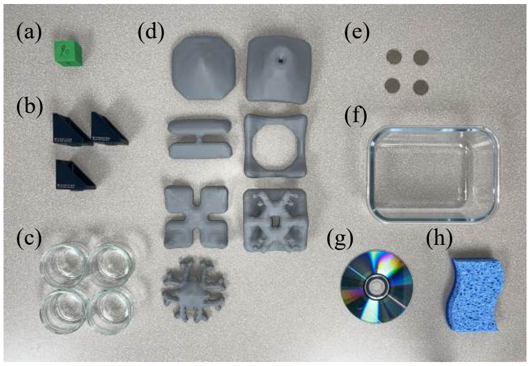

Fig. 6: **Objects transported in experiments.** (a) A 3D printed PLA cube: light objects with a smooth contact surface. (b) Aluminum gussets: heavy objects with a smooth contact surface. (c) Small glass containers: fragile objects. (d) EGAD! objects printed with PLA: large, irregularly-shaped objects. (e) US quarter coins: small, thin objects. (f) A large glass container: large, heavy, fragile objects. (g) A CD-R disc: large, thin objects. (h) A sponge: lightweight, deformable, and porous objects with an uneven contact surface.

We first measure μ_s for each material and the positions of objects' centroids when placed on the tray (either a single object at the center or multiple objects evenly around the center), then input these values into the optimizer to generate trajectories for each object. The optimizer sets the start and end position at (0.75, 0.30, 0.20) and (0.15, 0.80, 0.20) respectively, in meters relative to the robot base frame, and generates arced trajectories to move the tray. In contrast to the short, straight-line trajectories used during data collection, these longer arced trajectories test whether the constraint properly accounts for centrifugal and Coriolis forces. They also enable the robot to achieve higher acceleration within the same jerk limits, allowing us to evaluate the performance during faster transport. Additionally, maintaining the same start and end height lets us observe whether constraints with the learned friciton model allow the motion planner to optimize speed by tilting the tray [10].

For each object, we generate 1 time-optimized trajectory without constraints and 3 distinct constrained time-optimized trajectories: 1 with the standard Coulomb friction model (Coulomb), 2 with the learned friction model, trained with the PLA data (Learned-PLA) and the glass data (Learned-Glass) respectively. As the glass model is more conservative and results in a slower motion that takes t_{glass} seconds, we also generate 2 other trajectories for ablation study to verify the model is not just learning to go slower: 1 without constraints (AS-No Constraints) and 1 with the standard Coulomb friction constraint (AS-Coulomb), for both we terminate time optimization at t_{glass} seconds.

Experiments where objects fall off the tray are failure cases. On success, we measure object displacements relative to the tray in millimeters, by analyzing the top-down photos of the objects' initial and final positions with Segment

TABLE I: **Mean object displacement [mm] after transport.**

	Cube	Gusset	Glass small	EGAD!$^{(a)}$	Coin	Glass big	Gusset	Coin	Glass small	Disc	Sponge$^{(a)}$	Sponge$^{(a)}$
	×1	×1	×1	×1	×1	×1	×3	×4	×4	×1	hard side	soft side
Material	PLA	Aluminum	Glass	PLA	Cupronickel	Glass	Aluminum	Cupronickel	Glass	Polycarbonate	Polymer	Foam
μ_s	0.21	0.18	0.17	0.21	0.21	0.17	0.18	0.21	0.17	0.30	0.21	0.25
Coulomb	18.3	15.9	15.3	18.9	14.7	8.2	16.7	18.3	13.2	15.1	21.2	13.6
Learned-PLA	8.1	6.1	2.8	9.8	7.6	2.6	9.0	12.4	3.1	5.3	13.7	9.9
Learned-Glass	2.7	1.5	1.4	3.1	1.9	1.7	4.6	1.8	1.5	1.4	2.9	1.7
AS-Coulomb	3.9	2.7	2.1	4.4	3.2	2.3	4.6	3.6	2.3	3.8	5.8	1.1
AS-No Constraints	Fail	Fail	Fail	Fail	35.2	-*	Fail	26.5	Fail	21.2	38.5	15.6

Assumption violations: (a)uneven contact surface * Not tested for safety reasons.

TABLE II: **Time [s] to transport objects.**

	Cube	Gusset	Glass small	EGAD!	Coin	Glass big	Gusset	Coin	Glass small	Disc	Sponge	Sponge
	×1	×1	×1	×1	×1	×1	×3	×4	×4	×1	hard side	soft side
Coulomb	1.124	1.160	1.174	1.124	1.124	1.174	1.200	1.176	1.226	1.058	1.124	1.086
Learned-PLA	1.212	1.272	1.382	1.212	1.212	1.382	1.282	1.316	1.406	1.396	1.212	1.188
Learned-Glass, AS	1.406	1.500	1.516	1.406	1.406	1.516	1.526	1.500	1.525	1.372	1.406	1.376

Anything (SAM) [39]. In experiments involving multiple objects on the tray, the displacement of each trial is defined as the average displacement of all centroids in that trial. For the 7 EGAD! objects, we transport each of them once and compute the average displacement across the 7 trials. We transport each remaining object 5 times and take the average displacement from these 5 trials.

C. Results

Table I shows the average displacements in millimeters. Table II shows the duration of each trajectory in seconds.

All time-optimized trajectories without constraints result in failures, so we do not record them in the tables. Objects remain on the tray during transport with constraint using the standard Coulomb friction model, suggesting that the baseline constraint reflects some, but not all of the real-world effects. All objects transported with constraints using the learned friction model show improved performance, with object displacement reduced by an average of 53.9 % and 86.0 % with the PLA model and the more conservative glass model respectively. In ablation study AS-No Constraints, 6 out of the 11 tested objects slide off the tray, while the remaining exhibit significant displacement. Except for the 2 outliers, AS-Coulomb trajectories perform worse in all cases, with displacements up to 2.7× those from the learned constraint using the glass function. Among the 2 outliers, the gussets exhibit the same displacements using both trajectories, and the sponge is considered an out-of-distribution object due to its uneven contact surface: the hard side contacts only in the center, while the soft side touches at the edges. The ablation study indicates that the learned friction model captures phenomena beyond those accounted for by the standard Coulomb friction model, rather than scaling the measured friction coefficient by a constant $\alpha < 1$ to reduce transport speed.

For the comparison of speed, the learned friction model increases the transport time by an average of 12.6 % and 27.0 % using the PLA function and the glass function respectively. This suggests an opportunity for practitioners to make a trade-off between speed and displacement.

VII. Conclusion and Future Work

We propose learning a dynamic friction model with acoustic sensing to account for factors not characterized by the standard Coulomb friction model, and integrating it into a time-optimizing motion planner. We test on a UR5e robot with a tray as the end effector. Experimental results suggest that the learned friction model reduces object displacements after transport up to 86.0 % compared to constraining motions with the standard Coulomb friction model, making the proposed method practical in many real-world automation applications.

We consider 2 future directions for this work. First, the current tray design with a contact microphone is less effective at capturing vibrations from lightweight objects like a sponge. We will optimize the tray design and explore alternative piezoelectric sensors to improve sensing sensitivity. Second, we aim to extend friction modeling with acoustic sensing to more complex tasks, such as dexterous manipulation using higher-DOF end effectors.

References

[1] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, . T. Kröger, J. Kuffner, and K. Goldberg, "Dex-Net 1.0: A cloud-based network of 3D objects for robust grasp planning using a multi-armed bandit model with correlated rewards." IEEE, 2016, pp. 1957–1964.

[2] D. Wang, D. Tseng, P. Li, Y. Jiang, M. Guo, M. Danielczuk, J. Mahler, J. Ichnowski, and K. Goldberg, "Adversarial grasp objects," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 241–248.

[3] D. Morrison, P. Corke, and J. Leitner, "Egad! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation," 2020. [Online]. Available: https://arxiv.org/abs/2003.01314

[4] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-Net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning." IEEE, 2018, pp. 1–8.

[5] Y. Avigal, J. Ichnowski, M. Y. Cao, and K. Goldberg, "GOMP-ST: grasp optimized motion planning for suction transport," in *Algorithmic Foundations of Robotics XV - Proceedings of the Fifteenth Workshop on the Algorithmic Foundations of Robotics, WAFR 2022, College Park, MD, USA, 22-24 June, 2022*, ser. Springer Proceedings in Advanced Robotics, S. M. LaValle, J. M. O'Kane, M. W. Otte, D. Sadigh, and P. Tokekar, Eds., vol. 25. Springer, 2022, pp. 488–505. [Online]. Available: https://doi.org/10.1007/978-3-031-21090-7_29

[6] W. C. Agboh, J. Ichnowski, K. Goldberg, and M. R. Dogar, "Multi-object grasping in the plane," 2022. [Online]. Available: https://arxiv.org/abs/2206.00229

[7] W. C. Agboh, S. Sharma, K. Srinivas, M. Parulekar, G. Datta, T. Qiu, J. Ichnowski, E. Solowjow, M. Dogar, and K. Goldberg, "Learning to efficiently plan robust frictional multi-object grasps," 2023. [Online]. Available: https://arxiv.org/abs/2210.07420

[8] M. Selvaggio, A. Garg, F. Ruggiero, G. Oriolo, and B. Siciliano, "Non-prehensile object transportation via model predictive non-sliding manipulation control," *IEEE Transactions on Control Systems Technology*, pp. 1–14, 2023.

[9] A. Heins and A. P. Schoellig, "Keep it upright: Model predictive control for nonprehensile object transportation with obstacle avoidance on a mobile manipulator," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, p. 7986–7993, Dec. 2023. [Online]. Available: http://dx.doi.org/10.1109/LRA.2023.3324520

[10] J. D. Bernheisel and K. M. Lynch, "Stable transport of assemblies: Pushing stacked parts," *IEEE Transactions on Automation science and Engineering*, vol. 1, no. 2, pp. 163–168, 2004.

[11] F. Ruggiero, V. Lippiello, and B. Siciliano, "Nonprehensile dynamic manipulation: A survey," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1711–1718, 2018.

[12] P. Acharya, K.-D. Nguyen, H. M. La, D. Liu, and I.-M. Chen, "Nonprehensile manipulation: a trajectory-planning perspective," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 1, pp. 527–538, 2020.

[13] C. Zhou, M. Lei, L. Zhao, Z. Wang, and Y. Zheng, "Topp-mpc-based dual-arm dynamic collaborative manipulation for multi-object nonprehensile transportation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 999–1005.

[14] D. Gandhi, A. Gupta, and L. Pinto, "Swoosh! rattle! thump! – actions that sound," 2020.

[15] A. Thankaraj and L. Pinto, "That sounds right: Auditory self-supervision for dynamic robot manipulation," in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 229. PMLR, 06–09 Nov 2023, pp. 1036–1049. [Online]. Available: https://proceedings.mlr.press/v229/thankaraj23a.html

[16] Z. Liu, C. Chi, E. Cousineau, N. Kuppuswamy, B. Burchfiel, and S. Song, "Maniwav: Learning robot manipulation from in-the-wild audio-visual data," *arXiv preprint arXiv:2406.19464*, 2024.

[17] S. Clarke, T. Rhodes, C. G. Atkeson, and O. Kroemer, "Learning audio feedback for estimating amount and flow of granular material," in *Proceedings of The 2nd Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Billard, A. Dragan, J. Peters, and J. Morimoto, Eds., vol. 87. PMLR, 29–31 Oct 2018, pp. 529–550. [Online]. Available: https://proceedings.mlr.press/v87/clarke18a.html

[18] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 489–494.

[19] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 4569–4574.

[20] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization." in *Robotics: Science and Systems*, 2013, pp. 1–10.

[21] J. Ichnowski, M. Danielczuk, J. Xu, V. Satish, and K. Goldberg, "GOMP: Grasp-optimized motion planning for bin picking," in *2020 International Conference on Robotics and Automation (ICRA)*. IEEE, May 2020.

[22] J. Ichnowski, Y. Avigal, V. Satish, and K. Goldberg, "Deep learning can accelerate grasp-optimized motion planning," *Science Robotics*, vol. 5, no. 48, 2020.

[23] K. M. Lynch and M. T. Mason, "Dynamic underactuated nonprehensile manipulation," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96*, vol. 2. IEEE, 1996, pp. 889–896.

[24] ——, "Dynamic nonprehensile manipulation: Controllability, planning, and experiments," *The International Journal of Robotics Research*, vol. 18, no. 1, pp. 64–92, 1999.

[25] S. S. Srinivasa, M. A. Erdmann, and M. T. Mason, "Using projected dynamics to plan dynamic contact manipulation," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 3618–3623.

[26] J. Ichnowski, Y. Avigal, Y. Liu, and K. Goldberg, "Gomp-fit: Grasp-optimized motion planning for fast inertial transport," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 5255–5261.

[27] H. Pham and Q. Pham, "Critically fast pick-and-place with suction cups," *CoRR*, vol. abs/1809.03151, 2018. [Online]. Available: http://arxiv.org/abs/1809.03151

[28] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1307–1319, 2020.

[29] C. Wang, S. Wang, B. Romero, F. Veiga, and E. Adelson, "SwingBot: Learning Physical Features from In-hand Tactile Exploration for Dynamic Swing-up Manipulation," 2020.

[30] H. Zhang, J. Ichnowski, D. Seita, J. Wang, and K. Goldberg, "Robots of the Lost Arc: Learning to Dynamically Manipulate Fixed-Endpoint Ropes and Cables," 2021.

[31] C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Iterative residual policy: for goal-conditioned dynamic manipulation of deformable objects," 2022. [Online]. Available: https://arxiv.org/abs/2203.00663

[32] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. V. Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, N. Ratliff, and D. Fox, "curobo: Parallelized collision-free minimum-jerk robot motion generation," 2023.

[33] J. Y. Luh, M. W. Walker, and R. P. Paul, "On-line computational scheme for mechanical manipulators," 1980.

[34] L. Berscheid and T. Kröger, "Jerk-limited real-time trajectory generation with arbitrary target states," *Robotics: Science and Systems XVII*, 2021.

[35] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, 2015, pp. 18–25.

[36] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[37] T. Sainburg, "timsainb/noisereduce: v1.0," Jun. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3243139

[38] T. Sainburg, M. Thielk, and T. Q. Gentner, "Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires," *PLoS computational biology*, vol. 16, no. 10, p. e1008228, 2020.

[39] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.