AcoustoReinforce: Multi-Particle Acoustophoretic Path Planning with Deep Reinforcement Learning

Pengyuan Wei, Giorgos Christopoulos*, Zhouyang Shen*, Jincheng Wang, Joshua Mukherjee, Ryuji Hirayama, Sriram Subramanian, Prateek Mittal

Department of Computer Science, University College London
169 Euston Road, London NW1 2AE, United Kingdom
pengyuan.wei22@ucl.ac.uk

Abstract

Acoustophoresis uses sound waves to manipulate small objects in mid-air and has broad potential in various applications. However, stable multi-particle levitation remains challenging due to complex acoustic dynamics and limitations of existing models. We introduce AcoustoReinforce, a reinforcement learning-based path planner that autonomously controls the motion of multiple levitated particles. Leveraging a decentralized architecture, it learns local neural policies that generate particle trajectories independently, enabling scalable, communication-free control even in densely populated acoustic fields. To ensure physical feasibility, acoustic trapping strength is incorporated as a constraint during both training and inference, producing trajectories that are collision-free, acoustically stable, and physically realizable within real-world system constraints. Experiments on a real-world levitation platform show that AcoustoReinforce outperforms state-of-the-art planners, improving task success rates by up to 130% across diverse configurations. These results demonstrate the effectiveness of learning-based decentralized control for complex multi-object acoustophoresis in real environments.

Code and Appendix — https://github.com/pengyuanwei/AcousticLevitationEnvironment

Introduction

Acoustic holography enables various applications, such as contactless haptic (Shen et al. 2023, 2025; Mukherjee et al. 2025), and object levitation (Hirayama et al. 2019). Holographic acoustic levitation uses phased array transducers (PATs) to generate acoustic traps for manipulating particles (500 μm to 5 mm) in 3D space (Marzo et al. 2015). This capability supports applications in volumetric displays (Hirayama et al. 2019), additive manufacturing (Ezcurdia et al. 2022), and bio-manipulation (Ghanem et al. 2020). Multi-particle levitation can be viewed as an externally actuated multi-agent system, where each particle acts as an agent controlled by the PATs.

However, scalable control remains difficult. Phase-retrieval solvers (Hirayama et al. 2022) improve trap quality statically but often fail to maintain sufficient trapping force during motion. Recent planning methods (Gao et al. 2023) mainly optimize kinematics, ignoring acoustic constraints, which results in unstable trajectories and scalability issues as particle numbers increase. Therefore, path planning in acoustic levitation must address a fundamental physical challenge: the trapping force exerted on each particle must remain strong enough to keep it on its intended trajectory.

We propose AcoustoReinforce, a multi-agent reinforcement learning (MARL) planner that generates acoustically stable trajectories by explicitly incorporating trap quality. In addition to kinematic criteria, the method embeds acoustic trapping strength as a soft constraint in the learning objective. Specifically, it formulates multi-agent path planning as a decentralized problem and constrains trapping strength at turning points. We adopt a MARL algorithm within a centralized training and decentralized execution (CTDE) framework to train the policy (Lowe et al. 2017). During inference, AcoustoReinforce further improves solution quality by sampling candidate positions around each turning point and selecting those that yield more robust traps.

We validate AcoustoReinforce on a real-world, state-of-the-art levitation platform using unseen manipulation tasks. Compared to baseline methods, it consistently achieves higher particle manipulation stability and produces lower Gor'kov potentials at trajectory turning points across all tests. The main contributions of this work are:

- A MARL planner for multi-particle acoustophoretic manipulation that incorporates acoustic trapping strength as a soft acoustic constraint.
- A turning-point optimization module that further improves trapping strength by locally refining trajectories during inference.
- Real-system validation demonstrating substantial performance gains (up to 130% improvement over baselines) and stable control of up to 10 particles.

Background

Holographic Acoustic Levitation A phased array of N transducers (PAT) can generate different acoustic fields by controlling the amplitude and phase activations $a_n \in [0, 1]$, $\varphi_n \in [-\pi, \pi]$, $n = 0, \ldots, N$. For M target points, the

*These authors contributed equally.

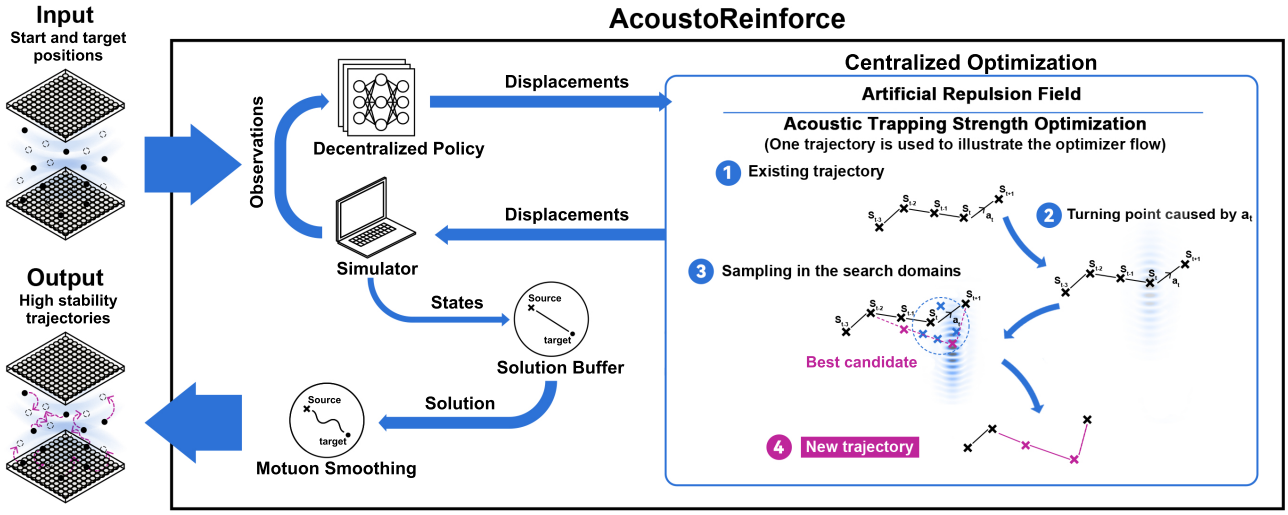

Figure 1: Overview of AcoustoReinforce: The process begins with the user specifying the initial and target particle positions, along with kinematic constraints. A decentralized neural network policy then generates collision-free trajectories for all particles. At each time step, the feasibility of the proposed displacements is evaluated. To maintain strong acoustic trapping at turning points, the acoustic optimization module refines the trajectory by randomly sampling new candidate positions near these points (indicated by the blue dashed circle in the figure) and selecting those that maximize acoustic quality.

resulting complex pressure field is given by the linear forward model $\boldsymbol{Th}$, where $\boldsymbol{h} = \boldsymbol{a} \odot e^{j\boldsymbol{\varphi}}$ is the acoustic hologram. $\boldsymbol{T} \in \mathbb{C}^{M \times N}$ is the transmission matrix that includes the individual source-to-point complex valued contributions, which is given by the piston model (Plasencia et al. 2020).

The acoustic radiation force on a small sphere is $\boldsymbol{F}_{rad} = -\nabla U$, where U is the Gor'kov potential (Marzo et al. 2015):

$$U = k_1|p|^2 + k_2\left(|p_x|^2 + |p_y|^2 + |p_z|^2\right).$$

Here, p is the acoustic pressure at the sphere location ($p = \boldsymbol{Th}$ for $M = 1$), and $p_{x,y,z}$ denote its spatial derivatives. The constants k_1 and k_2 depend on the sound speed and the material properties of the sphere. A sphere is levitated when the radiation forces converge toward it, and the point at which these forces converge from all directions is referred to as an acoustic trap (Marzo et al. 2015). For a strong trap, the divergence of the radiation force field must be minimized:

$$\nabla \cdot \boldsymbol{F}_{rad} = -\nabla^2 U,$$

where $\nabla^2 U$ is the Gor'kov Laplacian. However, computing $\nabla^2 U$ directly is expensive. Since the trap's U is strongly negatively correlated with its Gor'kov Laplacian (Hirayama et al. 2022), the potential U is commonly used as an efficient proxy for trap quality.

Multi-particle manipulation is typically achieved by repeatedly applying a time-invariant phase retrieval algorithm at each timestep along predefined trajectories. Phase retrieval algorithms compute the phases of PATs to generate acoustic traps. At each discrete time, $\boldsymbol{T}$ and $\boldsymbol{h}$ for the target points are computed. To ensure stable manipulation, it is essential to generate trajectories with high trap quality.

However, the complexity of the acoustic field makes path planning challenging. Since each transducer affects multiple particles, the resulting traps are inherently coupled. This coupling causes trap strength to be highly sensitive to spatial configurations and prone to interference, especially as particle count increases. As a result, the computational burden of planning grows nonlinearly with the number of particles.

Acoustophoretic Path Planning Mid-air multi-particle manipulation are typically relying on manually designed trajectories or conventional path planners. Optitrap (Paneva et al. 2022) introduces a trajectory optimization method to achieve more powerful shape rendering, but this method can only determine the optimal position and time for a single acoustic trap. DataLev achieves multi-particle manipulation by integrating the Scalable and Safe Multi-Agent Motion (S2M2) planning algorithm (Chen et al. 2021), which enables the generation of collision-free trajectories (Gao et al. 2023). StableLev employs an autoencoder model to detect anomalies in multi-particle trajectories and improves manipulation stability by reconstructing the acoustic field for unstable segments (Gao et al. 2024).

However, these methods fail to simultaneously satisfy three critical requirements: multi-particle manipulation, acoustic trap strength optimization, and real-time response. In contrast, deep reinforcement learning (DRL) offers advantages such as synchronized decision-making (Yang and Wang 2020) and efficient policy execution (Kulathunga 2022). AcoustoReinforce employs neural policies to accelerate path generation and leverages multi-agent synchronization to construct a sampling domain at trajectory turning points. This enables efficient search for candidate locations with higher acoustic trapping quality, thereby supporting real-time acoustic optimization. Experimental results show that AcoustoReinforce achieves more stable trajectories while maintaining high computational efficiency.

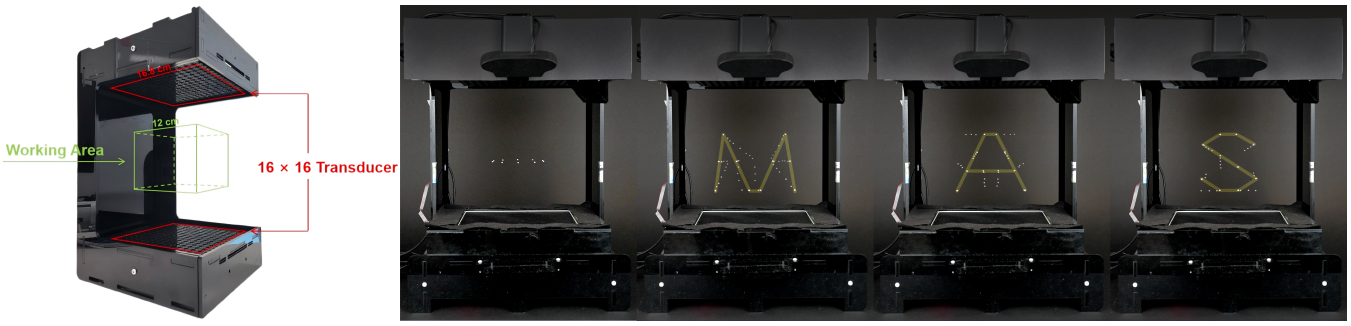

Figure 2: Left: A standard top-bottom setup levitator. Right: A demonstration of graphical transformation in an unseen task during model training: 10 levitated particles transition from a circular formation to target positions forming the letters "M," "A," and "S," with a maximum velocity of 0.1 m/s. The particle afterimages illustrate the trajectories generated by AcoustoReinforce, while the target positions are highlighted with light yellow lines. A video is provided as supplementary material.

Preliminaries

Multi-particle acoustophoretic manipulation can be formulated as an anonymous Multi-Agent Path Finding (MAPF) problem. Given N spherical particles with initial positions $\mathbf{x}_{src} = \{\mathbf{x}_{src}^1, \ldots, \mathbf{x}_{src}^N\}$ and target positions $\mathbf{x}_{tgt} = \{\mathbf{x}_{tgt}^1, \ldots, \mathbf{x}_{tgt}^N\}$, the Hungarian algorithm (Kuhn 1955) assigns targets optimally. The objective is to successfully transport all particles to their respective targets in $\mathbf{x}_{tgt}$.

The planning process is formulated as a finite discrete-time sequence $t = 0, 1, ..., T$, where T denotes the terminal time step when all particles reach their destinations. At each time step t, particle i ($1 \leq i \leq N$) is characterized by its current position $\mathbf{p}_t^i$ and target position $\mathbf{g}^i$. Given the system state, the planner calculates a displacement vector $\mathbf{d}_{t+1}^i$, and updates the particle's position to $\mathbf{p}_{t+1}^i$ for the subsequent time step $t+1$. The partial solution until time t is represented as $S_t = \{s_0, s_1, \ldots, s_t\}$, where s_t represents the complete system state at time t. The planner determines the subsequent positions of the particles based on s_t. To ensure stability, it must not only find collision-free paths but also ensure that the acoustic traps remain sufficiently strong at all times.

We use the Gor'kov potential U to evaluate the quality of the trap, with lower values indicating a stronger trap. Since the positions of all traps collectively determine the acoustic hologram, the displacements of all particles must be jointly optimized to ensure that the updated positions $\mathbf{p}_{t+1}$ satisfy the Gor'kov potential constraint $U_{\max}$. Consequently, multi-particle acoustophoretic path planning can be naturally framed as a sequential decision-making problem, where all particles share a common optimization objective: minimizing the expected average arrival time.

$$
\begin{aligned}
\underset{\pi}{\text{argmin}} \quad & \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} T_i \,\middle|\, \pi\right] \\
\text{subject to} \quad & \frac{(x_t^i - x_t^j)^2}{0.015^2} + \frac{(y_t^i - y_t^j)^2}{0.015^2} + \frac{(z_t^i - z_t^j)^2}{0.03^2} > 1, \quad \forall i, j, t \\
& \left\| \frac{\mathbf{d}_t^i}{\Delta t} \right\| \leq v_{\max}, \quad \forall i, t \\
& U_t^i \leq U_{\max}, \quad \forall i, t
\end{aligned}
$$

Here, π represents the path planning policy that determines displacement based on the system state. The index j refers to all particles except i. The velocity constraint is given by $v_{\max}$, and U_i^t denotes the Gor'kov potential at the position of particle i at time t. All constraints must hold at every timestep to ensure safe and effective manipulation.

Method

This section introduces our proposed acoustophoretic path planner, AcoustoReinforce (see Fig. 1). AcoustoReinforce employs a neural network policy trained using multi-agent reinforcement learning (MARL), which unfolds in two stages: offline policy training and online planning.

Policy Training

A. Reinforcement Learning Setup To derive the optimal policy π^* for the sequential decision problem introduced in Section Preliminaries, we formulate the task as a constrained decentralized partially observable Markov decision process (C-Dec-POMDP). This formulation explicitly incorporates the Gor'kov potential constraint into the optimization framework. The decentralized structure enables each agent to make independent decisions while maintaining a constant output size per actor, irrespective of the number of agents. This property significantly reduces policy learning complexity and ensures excellent scalability. The formal definition is provided below.

Definition 1 (Constrained Decentralized Partially Observable Markov Decision Process) A C-Dec-POMDP can be described as an 8-tuple:

$$\left\langle N, \mathbf{S}, \{\mathbf{A}^i\}_{i=1}^N, P, R, \{\mathbf{O}^i\}_{i=1}^N, \Omega, \mathcal{C} \right\rangle$$

- N: the number of agents.
- $\mathbf{S}$: the shared state space among all agents, with $s \in \mathbf{S}$.
- $\mathbf{A}^i$: the action space of agent i, with $a^i \in \mathbf{A}^i$. The joint action space is denoted as $\mathbb{A} := \mathbf{A}^1 \times \cdots \times \mathbf{A}^N$, and $\mathbf{a} := \{a^1, \ldots, a^N\}$ represents the joint actions.
- P: the state transition function, with $s' \sim P(s, \mathbf{a})$.
- R: the reward function shared by all agents, with $r^i = R^i(s, \mathbf{a})$ where $R^1 = \cdots = R^N$.

- $\mathbf{O}^i$: the observation space of agent i, where $o^i \in \mathbf{O}^i$. The joint observation space is $\mathbb{O} := \mathbf{O}^1 \times \cdots \times \mathbf{O}^N$, and $\mathbf{o} := \{o^1, \ldots, o^N\}$ represents the joint observations.
- Ω: the observation function, with $o^i \sim \Omega^i(s)$.
- $\mathcal{C}$: a set of constraint functions $\{C_k(s, \mathbf{a})\}_{k=1}^K$ with corresponding bounds d_k, such that each constraint satisfies:

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[C_k(s_t, \mathbf{a}_t)\right] \leq d_k.$$

The objective is to learn a policy π that maximizes expected cumulative reward while satisfying all constraints in $\mathcal{C}$. In our setting, one such constraint enforces the Gor'kov potential of each particle to remain below $U_{\max}$ at each step on average. The state space, observation space, action space and reward function of the problem are described in detail below.

1) State Space: the state s_t is defined as a vector of length $9 \times N$. It consists of the current positions $\mathbf{p}_t$, previous displacements $\mathbf{d}_t$, and target positions $\mathbf{g}$ of all particles:

$$s_t = [\mathbf{p}_t, \mathbf{d}_t, \mathbf{g}]$$

where

$$\mathbf{p}_t = [p_t^1, \ldots, p_t^N],\ \mathbf{d}_t = [d_t^1, \ldots, d_t^N],\ \mathbf{g} = [g^1, \ldots, g^N].$$

2) Observation Space: each agent's observation is a vector of length $3N + 6$, consisting of its current position $\mathbf{p}_t^i$, previous displacement $\mathbf{d}_t^i$, the relative position of its target point $\mathbf{g}^i - \mathbf{p}_t^i$, and the relative positions of all other agents $\mathbf{p}_t^j - \mathbf{p}_t^i$ for all $j \neq i$. The observation of agent i at time t is given by:

$$o_t^i = [\mathbf{p}_t^i, \mathbf{d}_t^i, \mathbf{g}^i - \mathbf{p}_t^i, \{\mathbf{p}_t^j - \mathbf{p}_t^i\}_{j \neq i}].$$

3) Action Space: each agent's action is represented as a displacement vector in a continuous space, specifying its movement along the x-, y-, and z-axes. Formally, the action of agent i at time step t is given by:

$$a_t^i = \mathbf{d}_{t+1}^i = [d_x^i, d_y^i, d_z^i].$$

4) Reward Design: As discussed in the Preliminaries, the objective of learning is to minimize the average time for particles to reach their target positions, subject to various constraints. The reward function for this cooperative task comprises a global reward R_G, a global cost, and a local coefficient R_L. The global reward and cost reflect system-wide goals, while the local coefficient ensures fair reward allocation among agents. The reward for particle i at each time step is defined as:

$$R(o^i) = \begin{cases} 20.0, & \text{if } d_g^j \in [0.0, 0.002), \forall i, \\ R_L(o^i) \cdot \sum_{j=0}^{N} \left[\text{Norm}(R_G(o^j)) + Cost(o^j)\right], & \text{otherwise.} \end{cases}$$

The global reward R_G is defined as:

$$R_G(o^j) = \begin{cases} 0.0, & \text{if collision occurs,} \\ \max\left(\frac{\mathbf{v}^j \cdot (\mathbf{g}^j - \mathbf{p}_t^j)}{\|\mathbf{g}^j - \mathbf{p}_t^j\|},\ 0.0\right), & \text{if } d_g^j \in [0.01, +\infty), \\ v_{\max} + 100(0.01 - d_g^j), & \text{if } d_g^j \in [0.002, 0.01), \\ v_{\max} + 0.8 + 200(0.002 - d_g^j), & \text{if } d_g^j \in [0.001, 0.002), \\ v_{\max} + 1.0, & \text{if } d_g^j \in [0.0, 0.001). \end{cases}$$

The global cost is given by:

$$\text{Cost}(o^j) = \min\left(\frac{0.2U_{\text{ref}} - U_j}{|U_{\text{ref}}|},\ 0.0\right)$$

The local coefficient R_L is defined as:

$$R_L(o^i) = \begin{cases} 1.0, & \text{if no collision,} \\ 0.0, & \text{if collision occurs.} \end{cases}$$

Here, U_{ref} denotes the mean of the Gor'kov potential distribution across the task.

B. Training Algorithm We adopt the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm (Lowe et al. 2017) to train our decentralized policies under the CTDE paradigm. The training follows standard MARL procedures for Dec-POMDPs with continuous action spaces. Detailed algorithmic settings are provided in the supplementary material.

C. Network Architecture All actor and critic networks are implemented as two-layer MLPs with 64 hidden units per layer. Each actor takes the agent's local observation as input and outputs a continuous action, while each critic evaluates the global state-action pair. Training is based on the Bellman loss and optimized using Adam.

Online Path Planning

The online phase iteratively alternates between generating synchronized linear segments for each agent using the offline-trained decentralized policy and refining them with centralized acoustic holography computations. The architecture of AcoustoReinforce is illustrated in Figure 1. The planning begins with the $\mathbf{x}_{src}$ and $\mathbf{x}_{tgt}$ of the suspended particles. This positional information is encoded by the simulator into an initial joint observation $\mathbf{o}_0$, which is then passed to the policy. Based on each particle's local observation o^i, the corresponding actor network predicts a straight-line displacement $\mathbf{d}_{t+1}$ for the next synchronized movement segment.

Before applying the displacements to update the system state, their feasibility is checked. If necessary, artificial repulsion fields and centralized acoustic optimizations are applied to refine the displacements. The simulator then updates the system state s with the refined $\mathbf{d}_{t+1}$. This iterative process continues until all particles reach their targets in $\mathbf{x}_{tgt}$. The complete path planning procedure is outlined in Algorithm 1, with its main components described below.

1) ISFEASIBLE(): Given a solution S_{t-1}, the current state s_t, and the joint action $\mathbf{a}_t$, this function checks whether all continuous paths from time 0 to time $t+1$, generated by extending S_{t-1} with the state-action pair $(s_t, \mathbf{a}_t)$, satisfy the distance constraint.

2) ARTIFICIALREPULSION(): Given that the performance of neural networks lacks theoretical guarantees, we introduce a time-invariant artificial repulsion field to correct states when the distance constraint is violated. The corrected positions matrix $\mathbf{P}$ is determined based on the step size η and the applied repulsion forces $\mathbf{F}$. The specific implementation

Algorithm 1: ACOUSTOREINFORCE($\mathbf{x}_{src}, \mathbf{x}_{tgt}$)

Input: $\mathbf{x}_{src}$, $\mathbf{x}_{tgt}$, policy $\boldsymbol{\pi}_\phi$, t_{max}
Output: trajectory S
s_0, $\mathbf{o}_0 \leftarrow$ INIT($\mathbf{x}_{src}, \mathbf{x}_{tgt}$), $S \leftarrow \{s_0\}$, $t \leftarrow 0$
while NOTREACHTARGET() $\wedge$ $t < t_{max}$ **do**
  $\mathbf{a}_t = \{a_t^1, a_t^2, \ldots, a_t^n\} \leftarrow \boldsymbol{\pi}_{\boldsymbol{\phi}}(\mathbf{o}_t)$
  if *not* ISFEASIBLE($S, s_t, \mathbf{a}_t$) **then**
    $\mathbf{a}_t \leftarrow$ ARTIFICIALREPULSION($S, s_t, \mathbf{a}_t$)
  end
  if DIRECTIONCHANGE($S, s_t, \mathbf{a}_t$) **then**
    $(s_t, \mathbf{a}_t) \leftarrow$ GORKOVOPT($S, s_t, \mathbf{a}_t$)
  end
  $s_{t+1} \leftarrow P(s_t, \mathbf{a}_t)$, $\mathbf{o}_{t+1} \leftarrow \Omega(s_{t+1})$
  S.APPEND(s_t)
  if TWGS() **then**
    RECORDPHASEREF()
  end
  $t \leftarrow t + 1$
end
$S \leftarrow$ MOTIONSMOOTHING(S)
return S

Algorithm 2: ARTIFICIALREPULSION($S, s_t, \mathbf{a}_t$)

Input: S, s_t, $\mathbf{a}_t$, step size η, max iters $T_{\max}$, sound field bound B
Output: $\mathbf{a}'_t$
$\mathbf{P} \leftarrow$ EXTRACTPOSITIONS($s_t, \mathbf{a}_t$), $\mathbf{P} \in \mathbb{R}^{N\times 3}$
for $k = 1$ **to** $T_{\max}$ **do**
  forces $\mathbf{F} \leftarrow \mathbf{0}_{N\times 3}$
  $\mathcal{E}(\mathbf{P}) = \{(i,j) : i < j,$ COLLIDE($\mathbf{P}[i], \mathbf{P}[j]$)$\}$
  for $(i,j) \in \mathcal{E}(\boldsymbol{P})$ **do**
    $\mathbf{f} \leftarrow$ REPULSEFORCE($\mathbf{P}[i], \mathbf{P}[j]$)
    $\mathbf{F}[i] \mathrel{+}= \mathbf{f}$, $\mathbf{F}[j] \mathrel{-}= \mathbf{f}$
  end
  $\mathbf{P} \leftarrow$ CLIP($\mathbf{P} + \eta\mathbf{F}$; B)
  $\mathbf{a}'_t \leftarrow$ UPDATEJOINTACTION($s_t, \mathbf{P}$)
  if ISFEASIBLE($S, s_t, \mathbf{a}'_t$) **then**
    return $\mathbf{a}'_t$
  end
end
return $\mathbf{a}'_t$

is detailed in Algorithm 2. At each iteration, the repulsion force between any two particles is given by:

$$\mathbf{f} = \mathbf{x} \odot \left[c \left(\frac{1}{\|\mathbf{v}\| + \epsilon} - \frac{1}{0.015} \right) \frac{1}{\|\mathbf{v}\|^2 + \epsilon} \hat{\mathbf{v}} \right],$$
$$\mathbf{v} = \frac{1}{\mathbf{x}} \odot (\mathbf{p}_i - \mathbf{p}_j).$$

where $\mathbf{x}$ is a scaling vector that adjusts the magnitude of the force, obtained by normalizing the safety distance along each direction. The $\mathbf{v}$ represents the scaled direction between the two particles, and the $\hat{\mathbf{v}}$ denotes the unit vector of $\mathbf{v}$. The ϵ is a small perturbation to avoid a zero denominator.

3) DIRECTIONCHANGE(): Determines whether the state-action pair $(s_t, \mathbf{a}_t)$ induces a significant direction change of any particle in s_t. A direction change is defined as an angular deviation greater than $2arctan(0.1) \approx 11°$, which filters out small oscillations in the policy output.

4) GORKOVOPT(): At each timestep, once DIRECTIONCHANGE() detects any turning points, the module performs a randomized optimization of them, as outlined in Algorithm 3. For each turning point, a spherical search domain is first constructed, and synchronous sampling is performed within these domains to generate a set of M candidate states $\mathbf{c} = \{s_t^1, \ldots, s_t^M\}$, while non-turning points remain fixed.

The qualities of the original state s_t and candidate states in $\mathbf{c}$ are then evaluated. As discussed in Section Preliminaries, the trapping strength of an acoustic trap can be characterized by its Gor'kov potential, with lower potential indicating stronger trapping. Consequently, the quality of a state is defined as the negative maximum Gor'kov potential among its turning points:

$$Q(s) = - \max_{\mathbf{p} \in \mathcal{T}(s)} \text{GORKOV}(s),$$

where GORKOV() is a function that calculates the Gor'kov potentials of all traps under a given state. $\mathcal{T}(s)$ denotes the set of turning points in state s. The candidates are then sorted by this quality measure, and the highest-ranked candidate replaces s_t if it satisfies feasibility constraints. This local refinement improves the trajectory stability (see Section Experimental Results) and introduces only a modest computational overhead, as quantified by the runtime statistics provided in Appendix.

5) CREATESEARCHDOMAIN(): Generates a spherical sampling domain D_i at each turning point $\mathbf{p}_t^i$:

$$D_i = \mathbb{B}(\mathbf{p}_t^i, R_i).$$

Let A_i be the circle centered at the midpoint between the two adjacent positions of p_t^i, with radius

$$r_i = \tfrac{1}{4} \|\mathbf{p}_{t-1}^i - \mathbf{p}_{t+1}^i\|.$$

If $\mathbf{p}_t^i$ lies inside A_i, the sampling radius R_i is set to the radius of the largest inscribed circle of A_i centered at $\mathbf{p}_t^i$. Otherwise, the sampling radius is simply r_i.

6) NOTREACHTARGET(): Checks whether all particles have reached their target positions in $\mathbf{x}_{tgt}$.

7) MOTIONSMOOTHING(): Applies S-curve or trapezoidal velocity smoothing to the initial and final segments of the motion.

8) GENERATECANDIDATES(): Performs random sampling within the search space to generate M candidate states.

9) RECORDPHASEREF(): Stores the holographic and target phases at state s_t as the reference for the TWGS solver.

Experimental Results

In this section, we evaluate the performance of AcoustoReinforce on both simulated and real levitation systems. The evaluation includes the success rate of the approach, the acoustic properties of the generated solutions, and the approach's success rate on a real levitation system under various configurations.

Algorithm 3: GORKOVOPT$(S, s_t, \mathbf{a}_t)$

$\mathbf{P} \leftarrow$ EXTRACTPOSITIONS$(s_t, \mathbf{a}_t)$
$Q_0 \leftarrow -\max_{\mathbf{p}\in\mathcal{T}(s_t)}$ GORKOV(s_t)
$D \leftarrow$ CREATESEARCHDOMAIN$(\mathcal{T}(s_t))$
$\mathbf{c} \leftarrow$ GENERATECANDIDATES(D)
for $s \in \boldsymbol{c}$ **do**
  $Q[s] \leftarrow -\max_{\mathbf{p}\in\mathcal{T}(s)}$ GORKOV(s)
end
for $c \in$ ORDERBYDESC$(\boldsymbol{c}, Q)$ **do**
  if $Q[c] \leq Q_0$ **then**
    break
  end
  $\mathbf{a}'_t \leftarrow$ UPDATEJOINTACTION$(c, \mathbf{P})$
  if ISFEASIBLE$(S, c, \mathbf{a}'_t)$ **then**
    if TWGS() **then**
      RECORDPHASEREF()
    end
    return $(c, \mathbf{a}'_t)$
  end
end
return $(s_t, \mathbf{a}_t)$

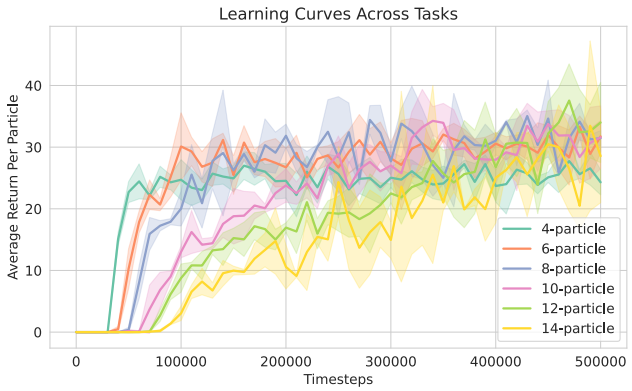

Figure 3: Learning curves of the policies trained four times per task. Shaded areas represent 95% confidence intervals.

Metric	Method	4	6	8	10
Success Rate	AcoustoReinforce	**0.9980**	**0.9980**	**0.9820**	**0.9830**
	S2M2	0.9810	0.9650	0.9280	0.8920
	CBS	0.9880	0.9890	0.9720	0.9370

Table 1: Success rate of different planners in simulation.

Training and Experimental Setup

The neural network policies were implemented and trained using PyTorch, with the MARL environment built on OpenAI Gym. Training was conducted on an Ubuntu 20.04 workstation equipped with an AMD Ryzen Threadripper 3960X CPU and an NVIDIA RTX 4080 GPU. For each specified number of agents, training took approximately 6 hours (corresponding to 5×10^5 time steps). The learning curves of the resulting policies are shown in Figure 3. Hyperparameter details for the MADDPG algorithm are provided in the supplementary material.

Experiments were conducted using an advanced acoustic levitation system with a top–bottom configuration of two planar PAT boards, a setup widely adopted in prior research, as shown in Figure 2 (Hirayama et al. 2019). The boards are horizontally aligned and spaced 23.4 cm apart. Each board contains a 16×16 grid of Murata MA40S4S transducers and is driven at 20 Vpp. The system is powered by the OpenMPD engine (Montano-Murillo, Hirayama, and Martinez Plasencia 2023). The transducers operate at 40 kHz with an update rate of 10 kHz. This setup forms an effective cubic working space measuring 12 cm per side, centered between the two boards.

In this work, we employed two hologram solvers: TWGS and Naive. For TWGS, the number of iterations was set to 5, the hologram phase change threshold to $\pi/32$, and the target point phase change threshold to 0 (Christopoulos et al. 2024).

Simulation-based Evaluation

We evaluate the performance of AcoustoReinforce and compare it with other state-of-the-art multi-agent path planning algorithms. AcoustoReinforce, S2M2, and CBS are applied to solve 1,000 randomly generated instances for each scenario, involving 4, 6, 8, and 10 levitated particles. Notably, AcoustoReinforce has not encountered these random instances during training. We did not evaluate the policy for 12 and 14 particles, as they exceed the physical limitations of the system. For S2M2, we adopt the same parameter settings used in DataLev (Gao et al. 2023). For CBS (Sharon et al. 2015), Space-Time A* (STA*) (Silver 2005) is employed as the low-level planner. To balance success rate and computational efficiency, the iteration limits are set to 20 for CBS and 100 for STA*.

Planning Success Rate. The performance results are presented in Table 1. Simulation results demonstrate that AcoustoReinforce achieves a consistently higher success rate ($98.3\% - 99.8\%$) compared to S2M2 ($89.2\% - 98.1\%$) and CBS ($93.7\% - 98.8\%$) across all scenarios. The advantage becomes particularly evident in more complex situations.

Acoustic Quality. We compare the Gor'kov potential at turning points for the solutions generated by each method. To evaluate the contribution of the policy and the optimizer, we conduct ablation experiments by generating solutions using only the policy, separated from AcoustoReinforce.

Figure 4(a) summarizes the results. As shown, the turning-point Gor'kov potentials of solutions generated by AcoustoReinforce are consistently lower than those of S2M2 and CBS, with reductions of up to 20%. As the number of particles increases (from 4 to 10), the Gor'kov potential exhibits a logarithmic-type decay, and the differences among path planning methods decrease. Without the optimization component, the policy improves the trapping quality at turning points by approximately 5%. In conclusion, the results demonstrate that solutions produced by AcoustoReinforce exhibit better acoustic performance-particularly when the number of particles is small-thereby validating the effectiveness of the acoustic optimization implemented by

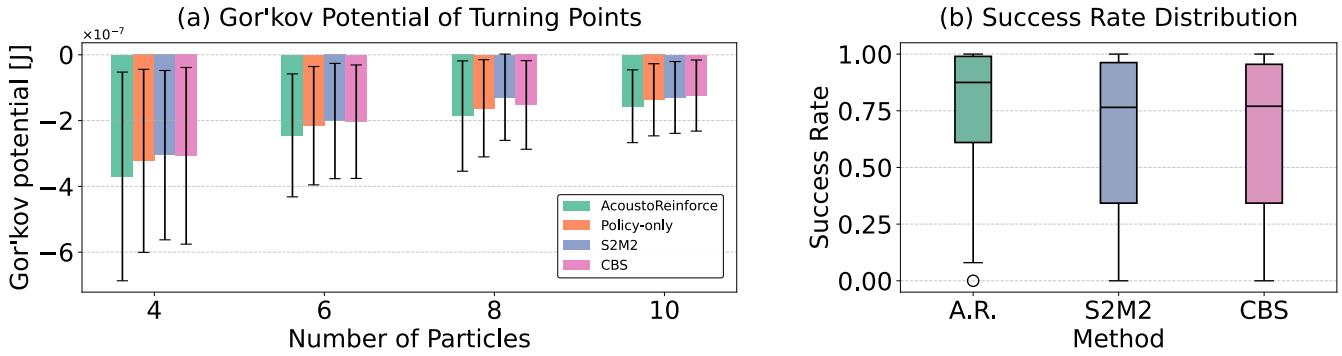

Figure 4: Performance comparison of path planners. (a) Gor'kov potentials at the trap center corresponding to the turning points, based on 4000 random samples per planner. (b) Distribution of real-world success rates for each planner across 16 configurations.

$V_{\max}$	Method	4	6	8	10
0.05 m/s	AcoustoReinforce	1.000	0.9900	**0.9900**	**0.6600**
	S2M2	1.000	0.9700	0.9600	0.4900
	CBS	1.000	**1.000**	0.9400	0.5200
0.10 m/s	AcoustoReinforce	1.000	**0.9400**	**0.9100**	**0.3400**
	S2M2	1.000	0.9200	0.8100	0.2300
	CBS	1.000	0.9000	0.7800	0.2400
0.15 m/s	AcoustoReinforce	1.000	**0.8400**	**0.7400**	**0.0800**
	S2M2	1.000	0.7900	0.6000	0.0000
	CBS	1.000	0.7600	0.5800	0.0100
0.20 m/s	AcoustoReinforce	**0.9200**	**0.6700**	**0.4600**	0.0000
	S2M2	0.7400	0.3800	0.2000	0.0000
	CBS	0.7900	0.3700	0.2600	0.0000

Table 2: Success rates of different methods on a real system with the TWGS solver.

AcoustoReinforce.

Real-System Experiment

Next, we validated our approach on a real-world levitation system, demonstrating that improvements in turning-point Gor'kov potential directly lead to higher solution success rates under identical kinematic constraints. To perform this, we generate 100 random instances for each scenario involving 4, 6, 8, and 10 particles. For each instance, we apply each path planning method to generate solutions at maximum velocities of 0.05, 0.1, 0.15, and 0.2 m/s. These solutions are then executed on the levitator, and the outcomes are recorded. The levitated objects are polystyrene beads with a diameter of 2 ± 0.5 mm. A test is considered unsuccessful if solution generation fails or if any particle falls during motion. Throughout the experiments, the device temperature was maintained below 38°C (Shen et al. 2024) to minimize the impact of temperature-induced pressure variations. The results are summarized in Table 2.

The experimental results demonstrate that AcoustoReinforce consistently enhances the stability of particle motion across all tested conditions, achieving success rates that match or surpass those of the baseline methods. Such enhancement could be attributed to the fact that AcoustoReinforce is the only method that directly optimizes acoustic parameters, while S2M2 and CBS focus only on resolving path conflicts. As shown in Table 2, when both the number of particles and the maximum speed (V_{max}) are low (e.g., 4 or 6 particles; 0.05 or 0.10 m/s), all three methods attain near-perfect stability. However, in more demanding scenarios (with higher speeds and large number of particles), the performance of S2M2 and CBS degrades rapidly, whereas AcoustoReinforce depicts robust success rates. For instance, with 8 particles and $V_{max} = 0.20$ m/s, AcoustoReinforce achieves a 46% stability rate—roughly double that of the baselines. Similarly, under lower-speed conditions with 10 particles, it yields up to a 50% improvement.

To further illustrate the robustness of each method, we visualized the success rate distribution using a box plot (see Figure 4(b)). The results reveal that AcoustoReinforce not only achieves a higher median success rate (approximately 0.915) but also exhibits a more concentrated distribution within the high-success region. This indicates greater consistency and reliability across varying scenarios. In contrast, S2M2 and CBS display more dispersed distributions, with a larger proportion of low-success samples—particularly under high particle counts and elevated speeds. These findings highlight the robustness and adaptability of AcoustoReinforce in challenging environments.

Conclusion

We propose AcoustoReinforce, a decentralized MARL-based path planning algorithm for stable, collision-free multi-particle acoustic levitation. By incorporating the Gor'kov potential into trajectory generation, it ensures both feasibility and acoustic quality. AcoustoReinforce achieves over a 90% success rate across various tasks and enables reliable control of up to 10 particles—outperforming existing methods that struggle under high particle counts. This approach enhances the robustness of acoustic manipulation and supports applications such as 3D displays, contactless objects transportation, and non-invasive medical procedures. Future work will explore real-world reinforcement learning for scenarios involving complex or unmodeled acoustic physics.

Acknowledgments

The authors thank the EPSRC Prosperity Partnership Programme - Swarm Spatial Sound Modulators (EP/V037846/1) - and the Royal Academy of Engineering, through their Chairs in Emerging Technology Programme (CIET 17/18), for their sponsorship, and Ana Marques for her support in creating the images.

References

Chen, J.; Li, J.; Fan, C.; and Williams, B. C. 2021. Scalable and safe multi-agent motion planning with nonlinear dynamics and bounded disturbances. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11237–11245.

Christopoulos, G.; Gao, L.; Plasencia, D. M.; Betcke, M.; Hirayama, R.; and Subramanian, S. 2024. Temporal acoustic point holography. In *ACM SIGGRAPH 2024 Conference Papers*, 1–11.

Ezcurdia, I.; Morales, R.; Andrade, M. A.; and Marzo, A. 2022. LeviPrint: Contactless Fabrication using Full Acoustic Trapping of Elongated Parts. In *ACM SIGGRAPH 2022 Conference Proceedings*, 1–9.

Gao, L.; Christopoulos, G.; Mittal, P.; Hirayama, R.; and Subramanian, S. 2024. StableLev: Data-Driven Stability Enhancement for Multi-Particle Acoustic Levitation. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1–11.

Gao, L.; Irani, P.; Subramanian, S.; Prabhakar, G.; Martinez Plasencia, D.; and Hirayama, R. 2023. DataLev: Mid-air Data Physicalisation Using Acoustic Levitation. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–14.

Ghanem, M. A.; Maxwell, A. D.; Wang, Y.-N.; Cunitz, B. W.; Khokhlova, V. A.; Sapozhnikov, O. A.; and Bailey, M. R. 2020. Noninvasive acoustic manipulation of objects in a living body. *Proceedings of the National Academy of Sciences*, 117(29): 16848–16855.

Hirayama, R.; Christopoulos, G.; Martinez Plasencia, D.; and Subramanian, S. 2022. High-speed acoustic holography with arbitrary scattering objects. *Science advances*, 8(24): eabn7614.

Hirayama, R.; Martinez Plasencia, D.; Masuda, N.; and Subramanian, S. 2019. A volumetric display for visual, tactile and audio presentation using acoustic trapping. *Nature*, 575(7782): 320–323.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2): 83–97.

Kulathunga, G. 2022. A reinforcement learning based path planning approach in 3D environment. *Procedia Computer Science*, 212: 152–160.

Lowe, R.; Wu, Y. I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30.

Marzo, A.; Seah, S. A.; Drinkwater, B. W.; Sahoo, D. R.; Long, B.; and Subramanian, S. 2015. Holographic acoustic elements for manipulation of levitated objects. *Nature communications*, 6(1): 8661.

Montano-Murillo, R.; Hirayama, R.; and Martinez Plasencia, D. 2023. OpenMPD: A low-level presentation engine for Multimodal Particle-based Displays. *ACM Transactions on Graphics*, 42(2): 1–13.

Mukherjee, J.; Shen, Z.; Christopoulos, G.; Subramanian, S.; and Hirayama, R. 2025. " To BEM or not to BEM?": Does Modelling Sound Scattering Improve Ultrasonic Mid-Air Haptics? In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 1–7.

Paneva, V.; Fleig, A.; Plasencia, D. M.; Faulwasser, T.; and Müller, J. 2022. OptiTrap: Optimal trap trajectories for acoustic levitation displays. *ACM Transactions on Graphics*, 41(5): 1–14.

Plasencia, D. M.; Hirayama, R.; Montano-Murillo, R.; and Subramanian, S. 2020. GS-PAT: high-speed multi-point sound-fields for phased arrays of transducers. *ACM Transactions on Graphics (TOG)*, 39(4): 138–1.

Sharon, G.; Stern, R.; Felner, A.; and Sturtevant, N. R. 2015. Conflict-based search for optimal multi-agent pathfinding. *Artificial intelligence*, 219: 40–66.

Shen, Z.; Bergström, J.; Vasudevan, M. K.; Obrist, M.; and Martinez Plasencia, D. 2025. Illusory-UMH: A Systematic Comparison of Tactile Illusions and Modulation Techniques in Ultrasonic Mid-air Haptics: Illusory-UMH. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, 1–13.

Shen, Z.; Morgan, Z.; Vasudevan, M. K.; Obrist, M.; and Martinez Plasencia, D. 2024. Controlled-STM: A Two-stage Model to Predict User's Perceived Intensity for Multi-point Spatiotemporal Modulation in Ultrasonic Mid-air Haptics. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1–12.

Shen, Z.; Vasudevan, M. K.; Kučera, J.; Obrist, M.; and Martinez Plasencia, D. 2023. Multi-point STM: effects of drawing speed and number of focal points on users' responses using ultrasonic mid-air haptics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–11.

Silver, D. 2005. Cooperative pathfinding. In *Proceedings of the aaai conference on artificial intelligence and interactive digital entertainment*, volume 1, 117–122.

Yang, Y.; and Wang, J. 2020. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv preprint arXiv:2011.00583*.