

Teaching Robots to Span the Space of Functional Expressive Motion

Arjun Sripathy, Andreea Bobu, Zhongyu Li, Koushil Sreenath, Daniel S. Brown, and Anca D. Dragan

Abstract—Our goal is to enable robots to perform functional tasks in emotive ways, be it in response to their users’ emotional states, or expressive of their confidence levels. Prior work has proposed learning independent cost functions from user feedback for each target emotion, so that the robot may optimize it alongside task and environment specific objectives for any situation it encounters. However, this approach is inefficient when modeling multiple emotions and unable to generalize to new ones. In this work, we leverage the fact that emotions are not independent of each other: they are related through a latent space of Valence-Arousal-Dominance (VAD). Our key idea is to learn a model for how trajectories map onto VAD with user labels. Considering the distance between a trajectory’s mapping and a target VAD allows this single model to represent cost functions for all emotions. As a result 1) all user feedback can contribute to learning about every emotion; 2) the robot can generate trajectories for any emotion in the space instead of only a few predefined ones; and 3) the robot can respond emotively to user-generated natural language by mapping it to a target VAD. We introduce a method that interactively learns to map trajectories to this latent space and test it in simulation and in a user study. In experiments, we use a simple vacuum robot as well as the Cassie biped.

I. INTRODUCTION

Robotics research tends to focus on generating functional motion, in service of the robot’s task. But imagine coming home from work exhausted and disappointed in being rejected from a job application, and the robot continuing to carry on its tasks as if nothing changed. Or, coming back with high energy after taking a walk in the sun, and the robot still putting the dishes away in the same exact way it always does. Ideally, our robots should adapt their behavior like in Fig. 1, including the way they move, in response to us, as well as in response to successes and failures they encounter, their confidence levels when performing a task, etc.

While much work has focused on expressive or emotive robot *gestures* [1]–[3], the ability to generate emotive *functional* motion that still achieves the robot’s task remains an open problem. How can a robot walk to its goal and avoid obstacles *while seeming happy or confident* in response to its user having had a great day? How could a manipulator place a dish in the sink while empathizing with its user’s disappointment at work? Taking an existing motion and adjusting its affective features, as researchers do with gestures [1], [4], would no longer meet the functional task specification. Instead, prior work [5] has proposed to learn a *cost function*

*This research is supported by the Office of Naval Research (ONR-YIP), the Air Force Office of Scientific Research (AFOSR), the UCSF Weill Institute for Neuroscience, and the Apple AI/ML PhD Fellowship. All authors are with the EECS department at UC Berkeley {arjunsripathy, abobu, zhongyuli, koushils, dsbrown, anca}@berkeley.edu

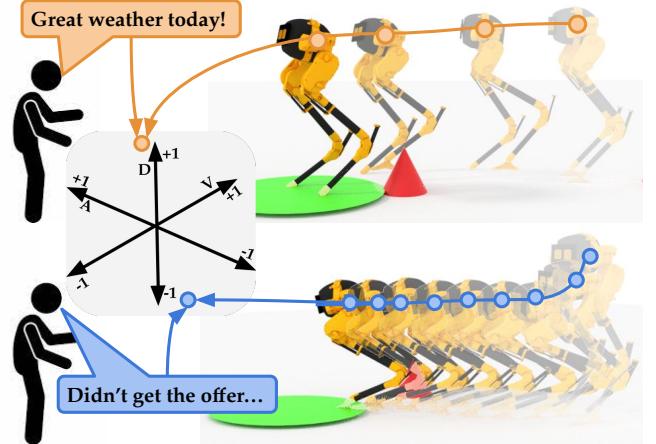


Fig. 1: Cassie robot performing a task with trajectories it believes exhibit similar emotion VAD as the human speech. (Top) The person’s happier sentiment (orange) results in the robot’s fast upright motion. (Bottom) The person’s sad sentiment (blue) results in slow and slouched motion.

from user feedback for each desired emotion or style, that can be then optimized along with the task specification. Although this addresses the problem of generating motion in an emotive style even as the specifics of the task change, it has the challenge that we have to think of every desired emotion in advance, and collect data specific to it. Further, we still need a way to decide which style or emotion to use.

In our work, we focus on the fact that emotions are not independent—they are latently related through the Valence-Arousal-Dominance (VAD) spectrum. Motivated by foundational studies in social psychology, VAD identifies three continuous, interpretable directions capturing much of emotional variance [6], [7]. Rather than learning independent cost functions for each emotion, our key idea is to learn to map robot trajectories to an emotive VAD latent space—this way, all user feedback contributes to learning about all emotions, and the robot can model new emotions that interpolate those seen during training. This enables robots to perform tasks in ways expressive of any specific emotion, by optimizing for a trajectory with a projection onto the latent space that is as close as possible to the desired emotion’s VAD. They also may use natural language to infer target emotion VAD: enabling stylistic response to emotive words, like “anger”, or even sentences, like “Great weather today!” as in Fig. 1.

Our approach interactively collects data from a user to learn this emotive latent space: it starts with an initial space, uses it to optimize emotive trajectories for a variety of task specifications and target emotions, asks the user to label these trajectories, and retrains the latent space to agree with the

user labels. Users may choose to label directly with VAD, or use language, which we can map to VAD by using pre-trained language models [8] finetuned to predict VAD scores.

In experiments with simulated human feedback for a Vacuum robot and the Cassie biped, we demonstrate the efficiency of our method in learning emotive costs when compared to approaches which model each emotion independently. We then show in a user study with the Vacuum robot that real humans can teach personalized emotive style in only 30 minutes of labeling. We find that users are able to recognize target emotions in robot motions generated with the model trained on their labels, even though those target emotions were not explicitly queried for during training. In summary, we propose a method for generating functional, stylistic robot motion by efficiently teaching how trajectories map to VAD defining a cost function encouraging target emotive style specified by natural language. Code and videos are made available at arjunsripathy.github.io/robot_emotive_space.

Despite showing promise in enabling simulated robots to perform functional tasks while expressing a wide range of emotions, much work remains ahead. Demonstrating generalization to a broader range of tasks than locomotion, thinking critically about how target VAD should be determined based on the emotion the user expresses, and moving as much of the process as possible to the physical domain all pose interesting challenges which we discuss further in Sec. V-B

II. RELATED WORK

Getting robots and virtual avatars to exhibit realistic looking and human-recognizable motions is a well-studied problem, from conveying intent in a task [4], [9], [10], to communicating incapability [11], [12], to expressing emotions [5], [13], [14]. In this section, we focus our attention on literature from the latter category, as our goal is enabling robots to learn emotive styles for performing functional tasks.

Motion style research has its roots in the graphics community. Some work looks at transferring motion capture style from one clip to another [15]–[17], but such unconstrained transfer is not appropriate for robots that need to satisfy rigid physical dynamics, or, even more challenging, to still be performing the desired underlying task. Alternative approaches use human demonstrations to learn locomotion styles as cost functions that the robot optimizes to respect task constraints [18], [19]. Unfortunately, due to the correspondence problem in robotics [20], these methods cannot be applied outside of locomotion robots, and acquiring demonstrations of stylized non-anthropomorphic robots is extremely challenging, especially when moving beyond gestures to functional motion.

In typical robotics motion style work, researchers design libraries of emotive motions that the robot can use during task execution [13], [14], [21]. To produce trajectories feasible for complex physical systems, Li et. al. [21] employ a dynamically constrained optimization that encourages the resulting motion to match stylized trajectories while abiding by the robot’s dynamics. The motions in these methods are

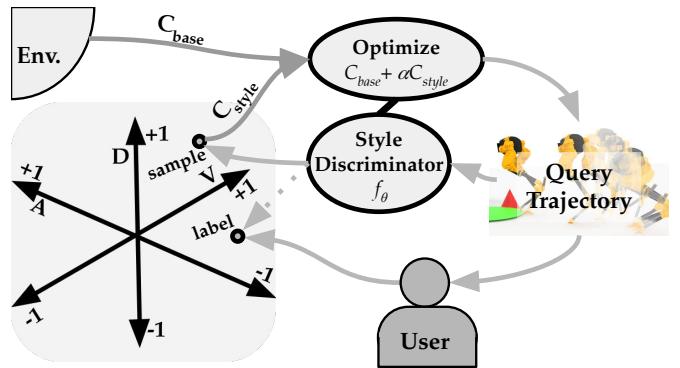


Fig. 2: Visualization of our method. The optimizer combines the task’s base cost C_{base} with the style cost C_{style} of a sampled emotion to produce a query trajectory the style discriminator f_θ believes aligns with the target VAD. The user labels the trajectory with a VAD, and the style discriminator f_θ is trained to bring its predicted VAD closer to the human label.

hand-crafted and, therefore, specific to the system and task they are being designed for. To generalize to a more diverse set of tasks, recent methods [5], [18], [19] try to learn a cost function that when optimized produces the desired emotive motion. However, these methods require collecting labels for each emotion one at a time, resulting in inefficient and costly learning that fails to generalize to new emotions.

Instead of representing each emotive motion with an individual cost function, we can learn a more generalizable representation. Sugitan et. al. [1] learned a latent emotive embedding along the VA spectrum [6], [7] capturing a whole space of emotions. While their approach enabled the robot to exhibit simple emotive gestures, like a slow lowering of the head for *sad*, we are interested in integrating emotive motion during the robot’s task execution. In this work, we take inspiration to similarly learn an embedding that maps emotive trajectories like the ones in Fig. 1 to a latent VAD space but extend their approach to functional task behaviors.

With this embedding, we have a representation that is both generalizable to new emotions and amenable to alternative forms of human feedback, such as natural language. Recently large, pre-trained language models such as BERT [8] have made transfer learning for downstream natural language tasks more accessible and efficient. Further, due to the breadth of research around VAD there exist datasets containing language and corresponding manual VAD annotations [22], [23]. Putting these together we may train a model for mapping natural language into our learned VAD space which will allow us to make the interaction between user and our system even more seamless.

III. METHOD

We now present our method for enabling a human to teach robots how to exhibit a broad range of emotions while performing various tasks. The core of our method is training a style discriminator, which predicts what emotion the human would perceive given a trajectory, using VAD labels collected from humans in response to query motions. For any target emotion, we’ll define a motion style cost based convincing

the discriminator that the trajectory being optimized exhibits the target VAD. We leverage the interpretable structure of VAD as a representation for emotion to improve learning efficiency, interpolate or extrapolate to new emotions, and integrate natural language seamlessly.

A. Preliminaries

We represent a trajectory $\xi \in \Xi$, where Ξ denotes the set of all trajectories in an environment, as a variable length sequence of waypoints along with the variable time duration between each pair of consecutive points. We denote environment tasks, such as moving from a specific start location to a specific goal, as $\tau \in \mathcal{T}$, where \mathcal{T} denotes the set of all tasks in the environment. The robot produces a trajectory that solves a particular task in the environment by optimizing a *base cost* $C_{base} : \Xi \times \mathcal{T} \rightarrow \mathbb{R}$. Optimizing a trajectory using C_{base} yields an efficient trajectory but offers no control over the emotion and typically produces a neutral style.

We describe the style of a trajectory based on an emotion VAD latent $e \in \mathcal{E} := [-1, 1]^3$, where the three values continuously represent Valence, Arousal, and Dominance in that order. Our goal is to learn a trajectory *style cost*, $C_{style} : \Xi \times \mathcal{E} \rightarrow \mathbb{R}$, capable of encouraging stylistic alignment with any target emotion e . Ultimately, to produce trajectories that achieve the task with the target style the robot will trade off between the base cost and the style cost:

$$C(\xi, \tau, e) = C_{base}(\xi, \tau) + \alpha \cdot C_{style}(\xi, e), \quad (1)$$

where α is a user specified hyperparameter that prioritizes between style and efficiency.

B. Cost Function Formulation

To learn the style cost C_{style} , we propose training a neural network style discriminator $f_\theta : \Xi \rightarrow \mathcal{E}$ parameterized by θ to map a trajectory ξ to the emotion e the robot exhibits while following it. Our motivation for this design is that every trajectory exhibits some emotion. The style discriminator f_θ naturally motivates a style cost function C_{style} which penalizes a trajectory ξ based on how much its exhibited emotion, $f_\theta(\xi)$, differs from the target e . We formalize this intuition using Euclidean distance in \mathcal{E} :

$$C_{style}(\xi, e) = \|f_\theta(\xi) - e\|_2^2. \quad (2)$$

By optimizing the combined cost function in Eq. (1) along with any task constraints, the resulting trajectory ξ completes the task while making its best effort to exhibit the target e .

C. Generating Human Queries

To train a robust discriminator f_θ , we generate batches of trajectories and query the user for emotive labels as shown in Fig. 2. The user provides either direct VAD labels or language which we map to VAD as discussed in Sec. III-E. Our goal is to learn θ , which is randomly initialized and updated after each labeling round as discussed in Sec. III-D.

To generate a round of query trajectories we optimize Eq. (1) for a batch of sample emotions and tasks, using the current estimate of θ for C_{style} . Given C_{style} does

VAD Emotive Latent Space

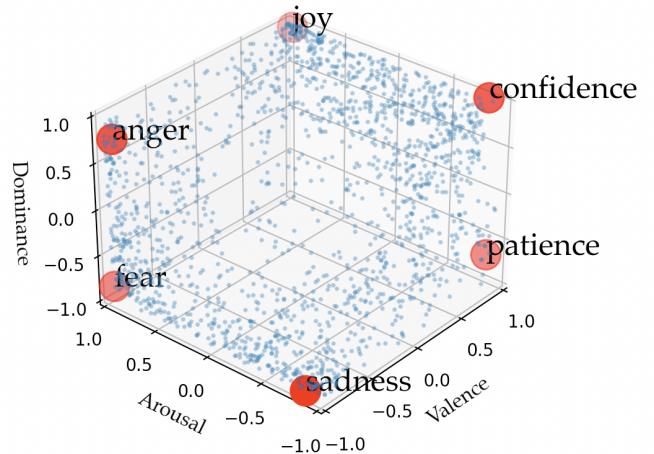


Fig. 3: The VAD latent space with each of Valence, Arousal, and Dominance being a real valued axis ranging from -1 to 1. The scatter plot depicts the projections of 1,672 emotive words projections into this space with the red dots highlighting the 6 basic evaluation emotions we used in our experiments.

not explicitly model the task, we focus on how to sample emotions in a way that is most informative for θ . Motivated by active learning literature we seek a diverse batch that biases towards *important*, unexplored areas of \mathcal{E} [24].

Not all areas of \mathcal{E} are equally important. Emotions are not uniformly spread across \mathcal{E} , and we would like to focus our queries on more populated areas of the space. We leverage the empirical emotion distribution from the NRC VAD lexicon [22], which contains annotated VAD values for 20k words. We filtered them down to 1,672 common emotive words, resulting in the VAD distribution in Fig. 3.

We now propose an active learning method for improving query coverage of this distribution to make the discriminator, f_θ , more robust. For the first round of queries, since the network f_θ is randomly initialized and has no semantic meaning yet, we uniformly sample B emotions from \mathcal{E} . To explain the process for successive rounds we must establish some notation. We conduct K query rounds with $k \in [1, K]$ referencing the round index. Let $\mathcal{S}_k := \{s_k\}^{1:B} \in \mathcal{E}^B$ reference the batch of sample emotions to be chosen by active learning to cover the empirical distribution of $D = 1672$ lexicon VAD values e_1, \dots, e_D . By optimizing Eq. 1 for \mathcal{S}_k alongside tasks randomly presented by environment, we generate query trajectories $\mathcal{Q}_k := \{q_k\}^{1:B} \in \Xi^B$. For these queries we will collect human labels referenced $\mathcal{L}_k := \{l_k\}^{1:B} \in \mathcal{E}^B$.

Our active learning method seeks to minimize the average distance between lexicon emotions and the closest acquired label from any round. This relies on estimating the \mathcal{L}_k based on our selection of \mathcal{S}_k . Our approximation here is $\mathcal{L}_k \approx \mathcal{S}_k$ which becomes more accurate over the course of training. With this assumption, for $k > 1$, we may select \mathcal{S}_k as:

$$\mathcal{S}_k = \arg \min_{\mathcal{S}_k} \sum_{i=1}^D \min_{l \in \mathcal{S}_k \cup \bigcup_{j=1}^{k-1} \mathcal{L}_j} \|e_i - l\|^2. \quad (3)$$

\mathcal{S}_k will bias towards densely populated areas of the VAD space where we do not yet have a nearby label. We approximate the optimal solution using expectation maximization [25]. We now turn our attention to how we may update θ after each round based on the collected human feedback.

D. Trajectory Network Training and Architecture

After each round k of querying, we update θ given our trajectory queries and VAD label responses collected so far ($\mathcal{Q}_{1:k}$, $\mathcal{L}_{1:k}$). We optimize the following MSE training loss:

$$L_k(\theta, \mathcal{Q}_{1:k}, \mathcal{L}_{1:k}) = \sum_{i=1}^k \sum_{j=1}^B \|f_\theta(q_i^j) - l_i^j\|_2^2. \quad (4)$$

Note the summand is exactly equivalent to $C_{style}(q_i^j, l_i^j)$ allowing for an alternative interpretation: we treat the queries as demonstrations for the emotion labels and would like to assign them minimal style cost.

In implementing this method, we have to choose a specific architecture for f_θ . Recall from Sec. III-A that we represent a trajectory as a variable-length sequence of waypoints and time deltas. We utilize an architecture similar to PointNet [26] for its simplicity and ability to gracefully handle varying length trajectories. First, a fully connected network processes each waypoint independently. Then we apply average and softmax pooling over waypoints to produce a single trajectory embedding. From there another fully connected network predicts the overall trajectory VAD value. Both networks use ELU activation [27].

It is important that the network predictions are smooth so they guide trajectory optimization well when used within the cost function. In other words, not only must predictions be accurate, but their gradient signal must also be informative. These factors motivated us to use smoother pooling (average & softmax) and activation functions (ELU), and to limit network capacity. A single hidden layer in each network, of dimensions chosen to match the complexity of the robot, along with L1 regularization worked well in our experiments.

E. Natural Language to VAD

We now describe how VAD may be inferred from natural language and where this may be used. Single words we look up directly in the NRC VAD lexicon [22]. For sentences, or words not present in the lexicon, we apply a BERT model finetuned to predict VAD using EmoBank: a dataset with 10k VAD labeled sentences [8], [23]. The wealth of resources and data around VAD is another benefit of using the spectrum as our latent space.

In many scenarios language provides a more natural means of communicating emotion than VAD. During training, language labels could be easier to provide compared to VAD directly. To be practical after training, the robot likely must determine target emotion in a less burdensome way than explicitly requesting VAD. Interpreting VAD from language allows the robot to seamlessly identify target emotion and modulate its behavior around humans.

IV. EXPERIMENTS

There are three primary hypotheses we seek to test with our experiments. (1) A real human is capable of using our method to teach their perception of a robot’s emotive style, and after training they perceive the robot’s intended target emotion in generated stylistic trajectories; (2) Our method is more efficient at learning a set of emotive styles than an approach that models each emotion independently; (3) Despite using general emotive labels, our method is equally efficient at learning any single emotive style as alternatives which leverage feedback specific to that emotion.

To evaluate hypothesis (2) & (3) we run a set of simulated human trials comparing the query efficiency our method to alternatives which model emotions independently. To test (1) we conduct a user study where real humans’ perception of emotion takes the place of simulated human heuristics. We evaluate the effectiveness of teaching by the extent to which the human (simulated or real) perceives emotion similar to what the robot intended to exhibit while completing various tasks. We will discuss the results for each case and ultimately find our hypotheses supported by the data collected.

A. Robots

We used two simulated robots to experiment with our method: a simpler Vacuum robot (VacuumBot) and the more complex and realistic Cassie bipedal robot [21]. We now describe the robot specifications, the environments they operated in, and the tasks they must complete.

VacuumBot is tasked with moving to collect dust that appears in a 2D world. It has 3 DoF controlling horizontal, vertical and angular acceleration and is subject to various physical constraints including gravity and friction. The current state of the robot and environment is summarized by the position and velocity corresponding to the DoF as well as the location of the dust. Trajectories are optimized for VacuumBot entirely using PyTorch [28].

Cassie, shown in Fig. 4, is a person-sized bipedal robot which has 20 DoF including 6 DoF of the base (its pelvis), 5 DoF and 2 passive joints of each leg. More details about Cassie can be found in [21]. In our paper, Cassie is tasked with navigating to a randomly generated target location while avoiding random obstacles represented by the red cones in Fig. 4. We leverage collocation to obtain a trajectory minimizing the proposed style cost while imposing constraints including collision-avoidance and reduced-order nonlinear dynamics as described in [29]. Based on the optimized trajectory of robot base velocity and height, the robot’s whole body motion is obtained from a gait library optimized by its full-order dynamics [21], [30]. This nonlinear optimization is formulated in CasADI [31], solved via IPOPT [32], and the resulting trajectory is visualized through animation in Blender [21]. To integrate our PyTorch implementation of f_θ we export our learned θ after every training round and replicate the neural network as a fixed numerical function in CasADI [31].



Fig. 4: Cassie must reach a target location avoiding obstacles represented by cones. Here it optimizes for various target emotions on various tasks using a model learned with our method. Visuals overlay one snapshot per second with earlier frames made more transparent. *Afraid* first takes a few cautious steps back before proceeding. *Happy* keeps head high and moves fast. *Sad* slouches and proceeds slowly.

B. Experimental Design

The simulated human trials and real human study used a very similar experimental design. In this section, we describe the process for evaluating a single method in general terms, and will discuss how this was adapted for the two contexts in their corresponding sections. We implement alternative models in a way that allows them to conform to the same evaluation procedure as our method.

Using notation from Sec. III-D, we train by conducting K rounds of B trajectory query batches updating the model after each batch of labels. Then we evaluate stylistic trajectories, produced for a representative set of evaluation emotions, based on the extent to which the human perceives the intended emotion in each.

To identify our evaluation emotions, we again leveraged VAD values from the NRC VAD Lexicon [22]. Based on the empirical distribution shown in Fig. 3, we identified the corners $[-1, -1, -1]$, $[1, 1, 1]$, $[-1, 1, -1]$, $[1, -1, 1]$, and $[-1, 1, 1]$ as the best regions to evaluate our model due to their population density and general coverage of the space. We selected representative emotions near each of these corners: *sadness*, *joy*, *fear*, *confidence*, *anger*, and *patience* respectively. Note that consecutive pairs on this list are diametric opposites in VAD space. We will not always use all 6 evaluation emotions and use N to reference the specific number we are working with. For consistency, when $N < 6$ we will always use the *first* N emotions based on the order we presented these above. We restrict ourselves to $N \in \{2, 4, 6\}$ to keep the emotions in opposing pairs.

Our three evaluation metrics are *Quality score*, *Top-1 accuracy*, and *Top-2 accuracy*. Quality score measures binary alignment: how well trajectories express the intended emotion compared to its diametric opposite. Top-X accuracies measure precise alignment: how well trajectories express the intended emotion compared to all $N - 1$ alternatives. Ideally we'd evaluate across all tasks, but given the task space is continuous we randomly sample M tasks for each of the N evaluation emotions averaging the metric values we get.

To compute the quality score, the robot presents the user with $N/2$ sets of $2 \cdot M$ trajectories for evaluation. Each set is associated with one diametric pair of evaluation emotions, say A and B. The $2 \cdot M$ trajectories includes M trajectories optimized for A and M trajectories optimized for B. The user is asked to assign each trajectory a score, s from 1 to 7 answering the Likert question: *Is the trajectory more*

expressive of Emotion B than A? with a response of 1 indicating the trajectory is very expressive of Emotion A and a 7 indicating that the trajectory is very expressive of B. Let q be the Quality score metric. For trajectories optimized for B we define $q := s$, and for A $q := 8 - s$. As a result, q ranges from 1 to 7 with 7 indicating perfect alignment with the intended emotion compared to its opposite.

To compute Top-X accuracies, we present the user with another $N \cdot M$ trajectories including M trajectories optimized for each of the evaluation emotions. Now the user is asked to select which of the N emotions is most expressed by each trajectory as well as their second choice. We define the Top-X accuracy metric, for $X \in \{1, 2\}$, as the proportion of the time the user's top X choices include the intended emotion.

C. Simulated Human Trials

We first conducted a set of experiments with simulated human feedback, since it would have been impractical to reliably test all our configurations with real humans.

1) *Simulating Human Feedback:* The simulated “human” (SH) uses heuristics to determine VAD for trajectories. For example, SH quantifies dominance for Cassie based on the average head height. SH may not accurately represent human emotive perception, but its consistent feedback allows us to compare learning efficiency between various methods.

During training SH directly provides its determined VAD for a trajectory as feedback. During evaluation, it must further transform this VAD value to mimic appropriate human responses. For the Likert question juxtaposing opposite emotions, SH projects the VAD value on to the diametric axis between the pair of emotions; then SH linearly scales the result so exactly Emotion A is 1 and exactly B is 7 clipping outside that range. For the choice based component, SH picks the closest and second closest evaluation emotion based on Euclidean distance in VAD space.

2) *Alternative Methods:* We compare our method to two alternative approaches which model emotions independently to test hypotheses (2) & (3). First is an approach inspired by Zhou et al. [5] which we'll reference as SEP, because it trains *separate* models directly predicting cost for each evaluation emotion. To do so, SEP directly asks the user to label trajectories with how expressive they are of *one* of the evaluation emotions. The second approach is SEP-ALL which we allow access to real valued cost labels for *all* evaluation emotions for each trajectory; SEP-ALL does not have to split its labeling budget between emotions as

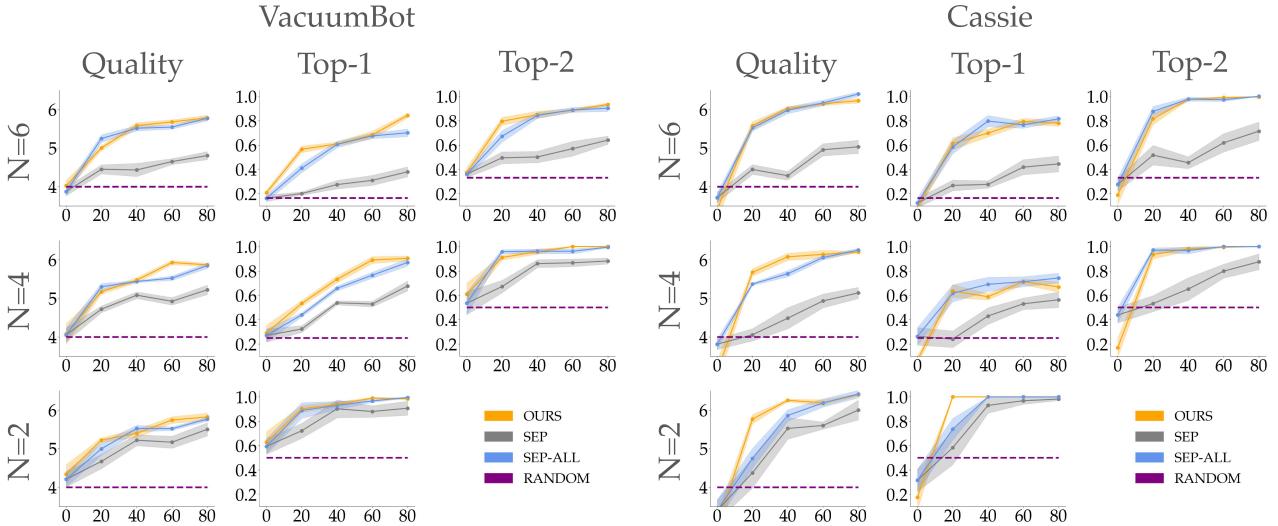


Fig. 5: Quality score, Top-1, Top-2 accuracy and standard errors over the course of training averaged across six seeds for each (environment, method, N) configuration. Metrics were computed before training and after each of 4 batches of 20 trajectory queries, with the query number indicated in the horizontal axis. Shown in dotted lines are the expected values if users chose randomly during evaluation. Our method is able to match performance of SEP-ALL whereas SEP struggles to keep pace as N , the number of evaluation emotions, increases.

SEP does. Recall our approach requests VAD labels from the user irrespective of the evaluation set enabling generalization beyond predefined emotions. In contrast, SEP and SEP-ALL both require knowledge of the evaluation set of emotions prior to training and get feedback specific to them.

To select emotions to generate trajectory queries for, SEP and SEP-ALL simply sample with replacement from the evaluation set. To remove a potential confound and isolate learning efficiency, we use the same selection process for our method in the simulated experiments. In the Sec. IV-D study we revert to the active learning described in Sec. III-C.

3) Simulation Results Discussion: The simulated human trials involved running each (environment, method, N) combination with 6 seeds, 108 experiments total, using $K = 4$ rounds of $B = 20$ trajectory labels. We present the average evaluation results using $M = 6$ tasks per emotion along with standard errors in Fig. 5.

To evaluate hypothesis (2), improving learning efficiency for a set of emotions compared to an approach that models them independently, we juxtapose our method with SEP. Across all metrics our method is able to reach a higher performance faster, the gap growing with N , supporting our hypothesis. With SEP each query is only informative for one of N emotive models. By contrast, with our method each query is informative for the entire VAD space and, thus, every evaluation emotion to some extent.

To evaluate hypothesis (3), matching learning efficiency for a single emotion compared to an approach that gets feedback specific to that emotion for each query, we juxtapose our method with SEP-ALL. SEP-ALL gets emotion specific cost labels for all N emotions with each query, as opposed to the generic VAD label our method receives. Yet across the board performance of our method matches SEP-ALL supporting our hypothesis again. It is not practical to go beyond a few evaluation emotions with SEP-ALL since labeling overhead scales linearly with respect to N , whereas

our method has constant overhead enabling capture of the full span of emotions. Furthermore, even for small N providing VAD values (or natural language) may be easier than real valued emotion specific costs.

Ultimately, VAD provides an interpretable latent representation that allows efficient learning of the space of emotive style with performance no worse than if we targeted any specific target emotion. Fig. 4 visualizes some emotive styles Cassie learned from this experiment, demonstrating our method’s ability to work with high DOF, complex robots.

D. Real Human Study

The benefit of this efficiency demonstrated in Sec. IV-C is that it becomes practical for real humans to use our system. In this section, we present a user study with VacuumBot aimed at testing hypothesis (1): the effectiveness of our method in teaching emotive style that is recognizable to end users.

1) Study Setup: We recruited 12 participants (9 male, 3 female) between 20 and 27 years old. They were asked to provide emotion labels for $K = 2$ rounds of $B = 20$ robot trajectories. We use a lower value for K here compared to the simulated experiments to emphasize the practicality of our approach. We found it was easier for humans to consistently label trajectories with VAD labels directly; however, as discussed earlier language may be used as well.

For the evaluation phase we used $N = 6$ emotions with $M = 3$ tasks each for all participants. To keep the overall study time shorter we did not perform intermediate evaluations, only evaluating after all labeling was complete. The labeling portion of the study took 30-40 minutes and the evaluation phase 20-25 minutes per participant.

2) Study Results Discussion: Fig. 6 compares human evaluation results to a random guess baseline. We break down results by each of the 6 evaluation emotions.

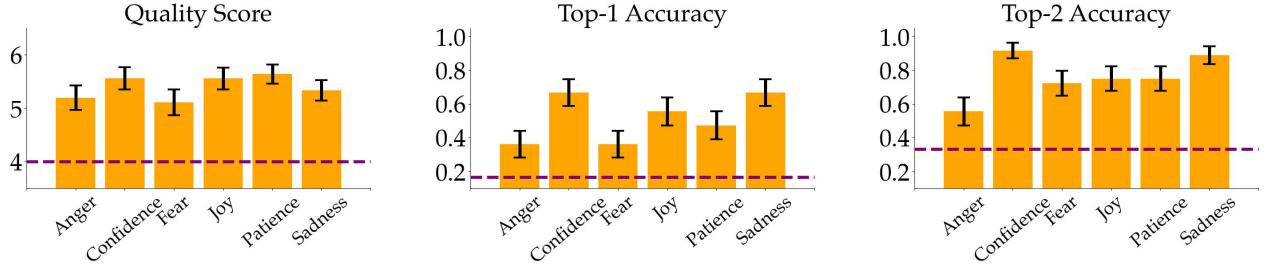


Fig. 6: Quality score, Top-1, and Top-2 accuracy and standard error for each evaluation emotion averaged across study users. Metrics significantly outperforms a random guess baseline, shown with the dashed lines, suggesting humans can indeed teach emotive motion with our method.

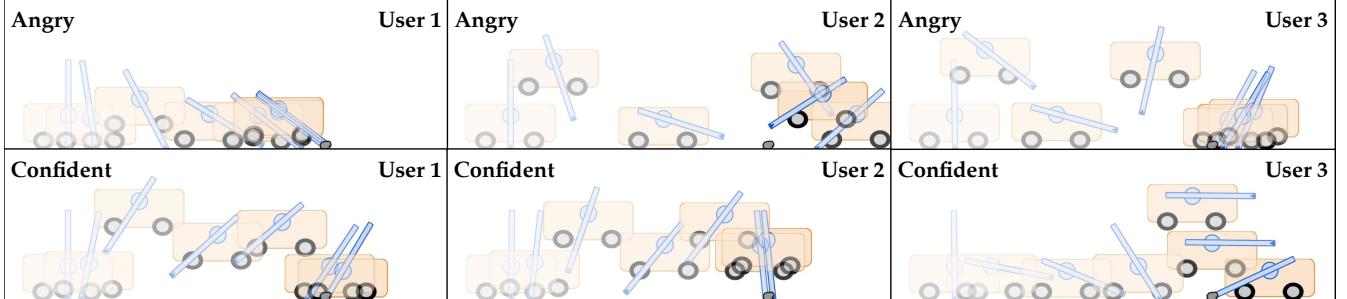


Fig. 7: VacuumBot collecting dust with style trained by three real users optimized for two target emotions on a single task. Judging by these motions, User 2 and 3's perception of anger involved greater speed, jumping, and arm movement than User 1. User 3's confidence had more arm movement but less jumping than User 1 and 2's.



Fig. 8: Three trajectories for a single task where VacuumBot infers VAD from the displayed phrases and produces motions based on a study user's trained model. When the human expresses cheerfulness it gracefully hops to the goal. When the human expresses sorrow it slowly slouches its way there. When the human expresses fear it reflects that nervous energy.

For every emotion and metric we ran t-tests at the 5% significance level comparing performance to the random baseline. Each test indicated statistically significant improvement which is reflected by the standard errors in Fig. 6. These results support hypothesis (1): our participants could teach the robot by labeling query trajectories in about 30 minutes, and during evaluation they perceived the robot's intended emotion at a rate significantly higher than random chance.

While the quality scores are relatively consistent across emotions the Top-1 accuracy varies a fair amount. This suggests the method is reliable in producing trajectories that are generally in the right direction but might not exactly line up with the intended one. In some use cases the former may be of primary importance and in others emotional precision may be equally important; however, it is reassuring to know that the robot will roughly align with the intended style even with it is not perfect in targeting the particular one.

A qualitative observation is that all users had their own personalized views of emotions. As demonstrated in Fig. 7, robots trained by three different study users ended up with fairly different, yet justifiable behaviors for the same emotions and task. Furthermore, in Fig. 8 we showcase example

motions for one user's robot generated based on the VAD of short phrases. This highlights one of the major benefits of our method: the robot learns more than a finite set of emotions from its teacher, instead learning an entire emotive space that it can index into and generate behavior with consistent results. By making the teaching process more accessible, our work takes an important step towards enabling anyone to teach robots nuanced behaviors without needing a technical foundation themselves.

V. DISCUSSION

A. Summary

We introduced a method that enables robots to perform functional tasks in ways that are expressive of a wide range of emotions. By teaching how trajectories map to VAD the robot may include a cost function encouraging a target emotive style in task motion optimization. Inferring VAD enables the robot to decide target emotions from human speech while in use and may also substitute VAD labels during training. Our experiments suggest that learning the VAD space jointly, beyond enabling emotion generalization, is more efficient and practical than trying to model each target

emotion separately. Furthermore, our experiments provide evidence that our method enables real humans to teach robots emotive style, to the extent where they can discern produced trajectories as expressive of intended emotions.

B. Limitations and Future Work

First we share some short term directions. Our environments only presented robot locomotion tasks, albeit varying the start, goal, and obstacle locations, hence expanding the task space to include more diverse objectives (e.g. object manipulation) would be an interesting direction. Although we mentioned the possibility of using language in place of VAD for training we did not explicitly evaluate this option. After training we propose inferring target VAD from user sentences, and despite promising qualitative results in Fig. 8 more in depth analysis is required. It's unclear even whether the robot should alter user emotion or merely reflect it.

Now we highlight some long term challenges. There are existing solutions for producing physical trajectories with our style cost [21], but bringing the training procedure into the physical domain is more challenging. It removes the ability to easily reset the robot as we do in simulation to facilitate label collection. Another challenge is while VAD captures the three most important emotive directions, sometimes differing emotions have similar VAD. For example, *fear* and *disgust* both have low valence and dominance with high arousal. Future work would have to navigate these subtleties while preserving the efficiency of our learning process.

We are excited about our results and believe they make an important contribution towards the end-goal of making robots more expressive and enabling people to teach personalized emotive styles. We look forward to seeing robots operate alongside humans with control over their exhibited emotion.

REFERENCES

- [1] M. Sugitan, R. Gomez, and G. Hoffman, “Demonstrating moveae: Modifying affective robot movements using classifying variational autoencoders,” *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020.
- [2] R. Desai, F. Anderson, J. Matejka, S. Coros, J. McCann, G. W. Fitzmaurice, and T. Grossman, “Geppetto: Enabling semantic design of expressive robot behaviors,” *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.
- [3] M. Saerbeck and C. Bartneck, “Perception of affect elicited by robot motion,” *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 53–60, 2010.
- [4] M. J. Gielniak and A. L. Thomaz, “Generating anticipation in robot motion,” in *2011 RO-MAN*, 2011, pp. 449–454.
- [5] A. Zhou and A. D. Dragan, “Cost functions for robot motion style,” *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3632–3639, 2018.
- [6] S. G. C.E. Osgood, *The measurement of meaning*. University of Illinois Press, 1957.
- [7] J. A. Russell, “A circumplex model of affect,” *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 12 1980.
- [8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *ArXiv*, vol. abs/1810.04805, 2019.
- [9] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, “Legibility and predictability of robot motion,” in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2013, pp. 301–308.
- [10] D. Szafir, B. Mutlu, and T. Fong, “Communication of intent in assistive free flyers,” ser. HRI ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 358–365.
- [11] M. Kwon, S. H. Huang, and A. D. Dragan, “Expressing robot incapability,” in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 87–95.
- [12] L. Takayama and D. Dooley, “Expressing thought: Improving robot readability with animation principles,” 01 2011, pp. 69–76.
- [13] H. Knight and R. Simmons, “Expressive motion with x, y and theta: Laban effort features for mobile robots,” in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, 2014, pp. 267–273.
- [14] M. Sharma, D. Hildebrandt, G. Newman, J. E. Young, and R. Eskicioglu, “Communicating affect via flight path: Exploring use of the laban effort system for designing affective locomotion paths,” in *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI ’13. IEEE Press, 2013, p. 293–300.
- [15] L. Torresani, P. Hackney, and C. Bregler, “Learning motion style synthesis from perceptual observations,” 01 2006, pp. 1393–1400.
- [16] S. Xia, C. Wang, J. Chai, and J. Hodgins, “Realtime style transfer for unlabeled heterogeneous human motion,” *ACM Trans. Graph.*, vol. 34, no. 4, jul 2015.
- [17] D. Holden, J. Saito, and T. Komura, “A deep learning framework for character motion synthesis and editing,” *ACM Trans. Graph.*, vol. 35, no. 4, jul 2016.
- [18] C. K. Liu, A. Hertzmann, and Z. Popović, “Learning physics-based motion style with nonlinear inverse optimization,” *ACM Trans. Graph.*, vol. 24, no. 3, p. 1071–1081, jul 2005.
- [19] S. J. Lee and Z. Popović, “Learning behavior styles with inverse reinforcement learning,” in *ACM SIGGRAPH 2010 Papers*, ser. SIGGRAPH ’10. New York, NY, USA: Association for Computing Machinery, 2010.
- [20] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.
- [21] Z. Li, C. Cummings, and K. Sreenath, “Animated cassie: A dynamic relatable robotic character,” *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3739–3746, 2020.
- [22] S. M. Mohammad, “Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words,” in *Proceedings of The Annual Conference of the Association for Computational Linguistics (ACL)*, Melbourne, Australia, 2018.
- [23] S. Buechel and U. Hahn, “Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis,” in *EACL*, 2017.
- [24] F. Zhdanov, “Diverse mini-batch active learning,” *ArXiv*, vol. abs/1901.05954, 2019.
- [25] F. Dellaert, “Expectation-maximization algorithm,” in *Encyclopedia of Machine Learning*, 2010.
- [26] C. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017.
- [27] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus),” *arXiv: Learning*, 2016.
- [28] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *NeurIPS*, 2019.
- [29] Z. Li, J. Zeng, S. Chen, and K. Sreenath, “Vision-aided autonomous navigation of bipedal robots in height-constrained environments,” *arXiv preprint arXiv:2109.05714*, 2021.
- [30] A. Hereid, O. Harib, R. Hartley, Y. Gong, and J. W. Grizzle, “Rapid trajectory optimization using c-frost with illustration on a cassie-series dynamic walking biped,” in *International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4722–4729.
- [31] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “Casadi: a software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, 2019.
- [32] L. T. Biegler and V. M. Zavala, “Large-scale nonlinear programming using ipopt: An integrating framework for enterprise-wide dynamic optimization,” *Computers & Chemical Engineering*, 2009.