2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

A Geometric Perspective on Visual Imitation Learning

Jun Jin†, Laura Petrich†, Masood Dehghan† and Martin Jagersand†

***Abstract*— We consider the problem of visual imitation learning without human kinesthetic teaching or teleoperation, nor access to an interactive reinforcement learning training environment. We present a geometric perspective to this problem where geometric feature correspondences are learned from one training video and used to execute tasks via visual servoing. Specifically, we propose VGS-IL (Visual Geometric Skill Imitation Learning), an end-to-end geometry-parameterized task concept inference method, to infer globally consistent geometric feature association rules from human demonstration video frames. We show that, instead of learning actions from image pixels, learning a geometry-parameterized task concept provides an explainable and invariant representation across demonstrator to imitator under various environmental settings. Moreover, such a task concept representation provides a direct link with geometric vision based controllers (e.g. visual servoing), allowing for efficient mapping of high-level task concepts to low-level robot actions.**

I. Introduction

Compared to traditional robotic task teaching methods, visual imitation learning promises a more intuitive way for general purpose task programming. Like most other learning methods, it suffers from the generalization problem. Commonly, three strategies are used to tackle generalization. The first one is to increase the number of human demonstrations via kinesthetic teaching or teleoperation. This has been proven effective for supervised learning methods such as behavior cloning [1]. However, it requires long and tedious human supervision work. The second strategy is to assume access to robot-environment interactions where more samples can be explored via reinforcement learning (RL) methods (e.g. IRL [2], GCL [3], GAIL [4]). Unfortunately, new issues regarding transfer learning and low sample efficiency arise during both simulation and real world training. The last strategy assumes that shared knowledge can be learned from demonstration samples across multiple but similar tasks; from this shared knowledge, the robot is able to learn a new task when given one more demonstration. This strategy is used in meta-learning based approaches (e.g. one-shot [5]).

Generally, aforementioned methods use human demonstration as *state-action* samples to learn a policy mapping from image to action (i.e. they approximate a target state-action distribution). Consequently, in order to improve generalization, it is necessary to collect more state-action experiences from either human teaching (supervision) or robot self-explorations (RL training). However, neither approach proves satisfactory. This motivates us to ask the question: *is it possible to learn by watching* ***one*** *human demonstration without the extra effort of interactive training?*

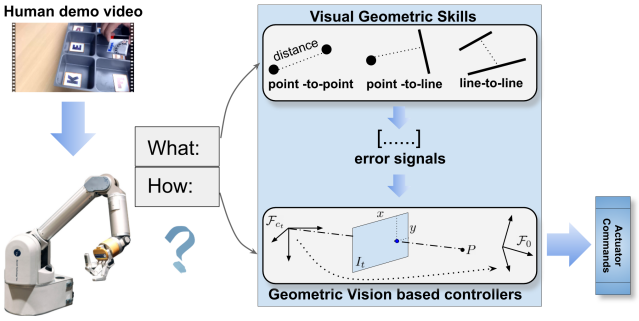

Fig. 1: Rethinking the classical 'correspondence' problem [6] in imitation learning reveals two essential questions: i) '*what*' information should be transferred from a human demonstrator to a robot imitator; and ii) '*how*' can this information be used to bring about actions (i.e. motor action). This paper presents a geometric perspective to this problem. We find a proper geometric representation of '*what*' that facilitates the training of '*how*'.

Recently, several methods have been proposed to tackle this question. One *key insight* is to rethink the classical 'correspondence' problem [6] which studies the difference between demonstrator and imitator. Such insight changes our view on human demonstration: what if more task concepts, instead of control, were encoded from the demonstrated data? Empirically, this aligns with our cognitive process in peer learning which involves first understanding the task before attempting any motor actions[1]. This hierarchical view decouples learning the 'what' and 'how' (zero-shot [8, 9], see Fig. 1). Benefits of this method are immediately observed: i) the promise to generalize well since it learns a high-level cognitive concept [10–12] of the task instead of directly matching state-action distributions; and ii) the promise of reusable low level polices as basic skills across different tasks [13]. However, two new problems arise: i) what is the high-level task; and ii) how can we train the low level controllers without an additional intensive cost.

In this paper, we provide a geometric perspective to derive solutions. We show that, instead of learning from image pixels to actions, learning a geometry-parameterized task concept[2] provides an explainable and invariant representation across demonstrator to imitator under various environmental settings. Moreover, it provides *controllability* that can be

†Authors are with Department of Computing Science, University of Alberta, Edmonton AB., Canada, T6G 2E8. {jjin5, laurapetrich, masood1, mj7}@ualberta.ca

[1]This is studied in observational learning [7] in psychology.

[2]For further reading, task parameterization using geometric constraints (e.g. point-to-point, point-to-line, point-to-conics, etc.) are intensively studied in [14–17].

directly linked to geometric vision based controllers (e.g. visual servoing). Our contributions are:

- We propose VGS-IL (Visual Geometric Skill Imitation Learning), an end-to-end geometry-parameterized task concept inference method used to infer globally consistent geometric association rules from demonstration video frames. Instead of learning from geometric primitives [13, 18] (e.g. points, lines, and conics) with handcrafted feature descriptors [19–21], VGS-IL can directly optimize a combinatorial representation from image pixels. Experiments show that the learned task concept generalizes well from human to robot despite the visual difference in arm and hand appearance.
- We show such geometry-parameterized task concept can be directly linked to geometric vision based controllers [22], thus forming an efficient way to map high-level task concept to low-level robot actions. Unlike prevalent methods requiring hierarchically training of an additional control policy [5, 8, 11, 23, 24], experimental results show that our learned representation fits directly into a visual servoing [25] controller, removing the need for feature trackers.

By using geometric primitive associations and 3D computer vision geometry based controllers, we present a method for general purpose robotic task programming.

II. Related Works

Visual Imitation Learning: The problem defined in visual imitation learning is: given one or several human demonstration videos, how can a new task be learned? Research on this topic dates back to 1994 [26, 27]. With the rise of deep learning and reinforcement learning, more influential works have since been published. While some are reviewed in section I, which aim to learn a task from visual inputs, it's worth noting another research stream aiming to learn a *semantic knowledge* representation. This method commonly relies on independent pipelines like object detection, action recognition etc. Despite of their method complexity, experiments show they can learn semantic task plans that follow a procedural manner [18, 28, 29].

Hierarchical Visual Imitation Learning: Instead of simultaneously learning task definition and control, hierarchical approaches decouple the two by focusing on learning a shared high-level task representation across human demonstrator and robot imitator. The two core problems are: i) how to represent the high-level task concept; and ii) how to train the low-level control policy. The first one is more important since representation of the task concept determines the controller training. For example, many pioneer works parameterize the task concept in *pixel level* by using sub-goal output from a neural network [8, 11]. The low-level policy is then sub-goal conditioned and trained following a Hierarchical Reinforcement Learning manner.

More recent works represent a task in the *object level* where object correspondence [23, 24] or graph structure [30] relationships are utilized to parameterize a task. The low-level controller is then trained based on distance errors in the embedded parameterization space. This approach shows success in pushing and placing tasks, however, it lacks the definition resolution required for more complex tasks like insertion.

Geometry-Based Visual Imitation Learning: Alternatively, going deeper inside the objects, *geometric feature level* based approaches arise. Early pioneer works from Ahmadzadeh et al. 2015 [18] proposed VSL to learn feature point correspondence based task representation given one human demo video. A similar approach from Qin et al. 2019 [31] presents KETO which utilizes key point relationships to represent a tool manipulation task. In general, their low-level controllers are tediously trained separately without enough study emphasis on how a proper task representation will facilitate the low-level policy training.

Beyond a simple key point correspondence based task concept representation, other basic geometric constraints (point-to-line, line-to-line, etc.) can enrich our toolbox for parameterization of task concepts [32]. Furthermore, by concurrently combining and sequentially linking them [16], we can find a general way to program more complex manipulation tasks that exhibit scalability. To the authors best knowledge, applying such systematic geometry-based task programming in visual imitation learning is rarely studied.

III. Method

This research builds upon our previous work on visual geometric skills learning [13], while employing a more data driven approach to learn globally consistent geometric feature association rules without hand crafted feature descriptors [19–21]. The basic idea of representing a manipulation task using geometric feature association rules, as firstly proposed in [14], has two parts: i) the basic geometry constraints or visual geometric skill (VGS) kernels [13]; and ii) the combination or conditioned linking of basic kernels to create more complex tasks, which we refer to as visual geometric skills. For example, some commonly used geometric skill kernels are:

- *point-to-point* gk_{p2p}: the coincidence of two points.
- *point-to-line* gk_{p2l}: a point fits on (touches) a line.
- *line-to-line* gk_{l2l}: a line is co-linear with another line.
- *point-to-conic* gk_{p2c}: a point fits a conic.

This task representation method provides a programmable framework that can be used to create more complex tasks by combining several constraints in parallel and then sequentially linking the basic kernels. For example, inserting a pen tip inside its cap involves a point-to-point constraint linked by a line-to-line one (Fig. 2A).

A. Geometry parameterized task representation

Firstly, *we show how to parameterize a VGS kernel.* Given a set of geometric features $\{f_i\}$, a VGS kernel is an operator gk that maps $\{f_i\}$ to a latent vector which is computed using geometric constraints from $\{f_i\}$ to measure the error signals related to the task goal. We propose three essential properties of gk:

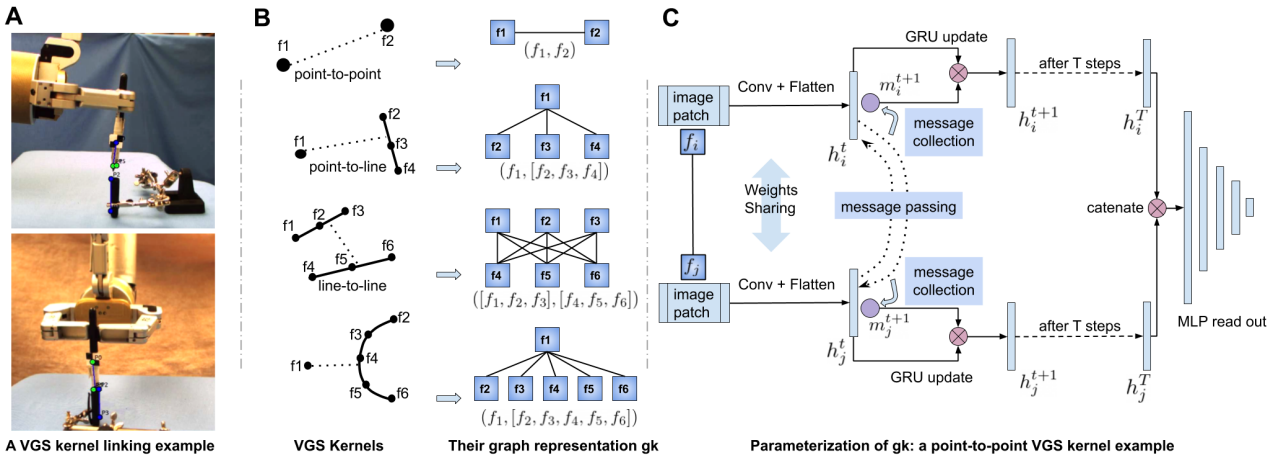

Fig. 2: **A**: An example of visual geometric skill (VGS) kernel linking task [32]: Inserting a pen inside its cap involves a point-to-point constraint (pen tip touches cap top) followed by a line-to-line constraint (align pen direction to the cap). **B**: Basic geometric skill kernel representations using a graph structure. More complex tasks are programmed by combining the basic kernels. **C**: An example of parameterization design of the point-to-point kernel gk_{p2p}. Using a message passing graph neural network with a GRU update fulfills the three required properties for a good gk representation. Other visual geometric skill kernels can be parameterized in a similar manner.

- *Communicative:* A good operator gk should be consistent for all input features sequence orders. For example, when we enumerate all possible associations of three features, we require $gk(f_1, f_2, f_3) = ... = gk(f_3, f_2, f_1)$ for all 6 possible permutations since they define the same task.
- *Non-inner-associative*: A good gk should be able to represent $gk(f_1, [f_2, f_3, f_4]) \neq gk([f_1, f_2, f_3], f_4)$. For example, consider a point f_1 and a line defined by three points (f_2, f_3, f_4). A point-to-line kernel operation is $gk(f_1, [f_2, f_3, f_4])$, which is unique from any other inner associations.
- *Scalability:* The ways to parameterize gk should be scalable to fit n-ary operations. For example, a point-to-point kernel is a binary operation while point-to-line is a quaternary operation if three points represent a line.

Examples of basic VGS kernel parameterizations are included in Fig. 2B. As shown in Jun et. al [13], a parameterization is found using a message passing graph neural network [33] with a gated recurrent unit GRU [34] that satisfies the above properties: i) a graph structure is scalable to represent n-ary operations; ii) graph edges define different inner associations; and iii) the message passing mechanism combined with GRU makes gk invariant from input orders. Specifically, this design (Fig. 2C) has four steps: A pair-wise message generation $\mathcal{M}$:

$$m_{i\to j}^{t+1} = \mathcal{M}(h_i^t, h_j^t) \tag{1}$$

, where h_j^t, h_j^t are connected nodes' hidden states. A message aggregation $\mathcal{A}$ which collects all incoming messages:

$$m_i^{t+1} = \mathcal{A}(m_{i\to j}^{t+1}) \tag{2}$$

A message update $\mathcal{U}$ using a gated recurrent unit (GRU):

$$h_i^{t+1} = \mathcal{U}(h_i^t, m_i^{t+1}) \tag{3}$$

Finally a readout function is parameterized using MLP layers. After T layer updates, all nodes' final states are fed into a readout function: $b = \text{MLP}(h_1^T, ..., h_n^T)$.

Next, we show *how to encode the graph entities.* Previous works [13, 35] used hand crafted feature descriptors in training. Is it possible to utilize the representation capability of deep learning by directly learning from raw images? Moreover, for simple geometric features like points and lines, there are on-the-shelf descriptors that can be used. What about more complex geometric primitive like conics and planes? In this paper, we propose a composable graph structure used to encode geometric primitives with point-based image patches, as shown in Fig. 2B.

Lastly, a visual geometric skill (VGS) is composed by combining or linking multiple VGS kernels. This paper will only cover kernel combinations and will leave kernel linking for future research.

B. VGS learning by watching human demonstrations

Assume a VGS task consists of multiple geometric skill kernels $\{gk_i\}$, learning VGS becomes the optimization of each gk_i given a human demonstration image sequence $\{I_t\}$.

1) Select-out function.: An optimal gk_i selects the *right* geometric feature associations out of a set of combinatorial instances. For example, in the point-to-point kernel, we can get N feature points from one image by applying any feature extractor. To enumerate, there are $m = C_N^2$ candidate instances. Suppose each instance has an output b_j by applying the operator gk_{p2p}. We compute its relevant factor $g_j = \text{softmax}(b_j, \{b_1, ..., b_m\})$ and the right one is selected out from the maximum g_j.

2) Optimization: Applying gk_i on each image frame I_t will output a control error signal $\mathbf{e}_t$[3]. Assuming gk_i is optimal, applying gk_i on a human demonstration image

[3]For example, a point-to-point kernel outputs x-y errors in image pixels. A point-to-line kernel outputs error signal from the dot product of their homogeneous coordinates. More examples can be found in [36].

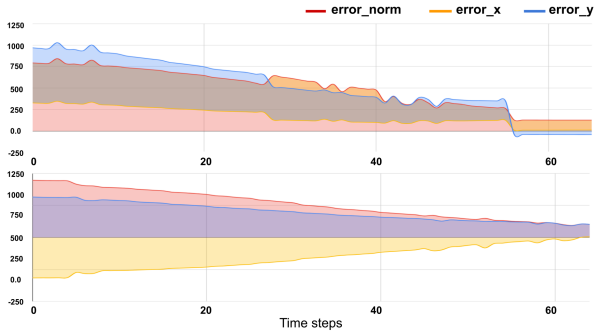

Fig. 3: Given human demonstration video frames, optimizing gk_i is essentially selecting the right observation space to measure a high-quality control error signal output from the human expert demonstrator. By maximizing the quality of control error signals, we adjust our $\{gk_i\}$ estimation. **Up**: The control error output of gk_i trained without geometry consistent regularizer (GSR) of the *sorting* task described in Sec. IV. **Down**: The control error output of a well-trained gk_i by adding GSR to the loss function.

sequence $\{I_t\}$ will output a high-quality control error signal sequence $\{\mathbf{e}_t\}$. Hence, optimizing gk_i is essentially selecting the right observation space to observe a high-quality control error signal output from the human demonstrator. We call this the ***observational expert*** assumption. By maximizing the quality of control error signals, we are able to adjust our gk_i estimation (Fig. 3).

We measure the quality of control signals by a reward function using two metrics: i) errors are overall decreasing along the time steps of human demonstration; and ii) error changes are smooth. The first metric is encoded into the reward function as defined in [13]. To achieve smoothness, we modify the loss function defined in [13] by adding a geometry consistent regularizer (GCR): $-\alpha \left\| b_{t+1} - b_t \right\|_2^2$ while keeping the same residual sum of weights (RSW) regularizer for deterministic selection purpose. GCR forces learning a more consistent selection across frames.

By optimizing the reward function using InMaxEntIRL [**?**, 13], the control signal quality from the human demonstrator is optimized, resulting in an optimized gk_i. To summarize, we propose VGS-IL (Visual Geometric Skill Imitation Learning) as detailed in Algorithm 1.

C. Links to geometric vision-based controllers

The control signal $\mathbf{e}_t$ output from gk is observed in image pixel space. Mapping image observations to robot actions is a long running research topic [37] also known as robot eye-hand coordination, visuomotor policy learning, or vision guided robot control [22]. Approaches can be divided into two categories[4]: i) end-to-end learning methods [39]; and ii) visual servoing (VS) [25]. End-to-end learning approaches can work without explicit features, and are useful in complex visual environments due to their powerful representation capability [40], but require time consuming training and show poor transfer to new environments (i.e. poor generalization). Visual servoing approaches run in real-time using a

[4]For further reading, a comparison has been discussed in the ICRA 2018 Tutorial on Vision-based Robot Control [38]

Algorithm 1: VGS-IL

Input: Expert demonstration video frames $\{I_1, ..., I_n\}$, demonstrator confidence level α, VGS=$\{gk_1, ..., gk_m\}$
Result: Optimal weights $\boldsymbol{\theta}_i^*$ of gk_i
Construct kernel graph instances on each frame
for *i=1:m* **do**
  Define $\mathcal{S} = \{\}$
  for *t = 1:n* **do**
    Feature extraction on I_t according to gk_i defined in Section III-A
    $s_t \leftarrow$ Construct all gk_i instances by feature association
    $\mathcal{S} \leftarrow$ Append s_t
  end
  Prepare State Change Samples
  $\mathcal{D}s = \{s_t \rightarrow s_{t+1}\}$
  $\boldsymbol{\theta}_i^*$ = InMaxEntIRL($\mathcal{D}s, \alpha, gk_i$) [13]
end

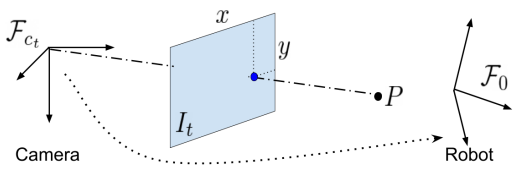

Fig. 4: A geometric vision-based controller utilizes camera 3D geometry to build relationships between 3D object motion, observed feature motion in image plane I_t and camera spatial velocity. At last it links with robot actions via a calibration model or trial-error manner based online learning.

geometric vision-based control law, but can lack sufficient visual representation capability. Combining the geometric vision part from visual servoing with learning-based methods is rarely studied [41, 42].

As shown in Fig. 4, the basic idea of using geometric vision in VS control is: i) mapping an error vector $\dot{\mathbf{e}}_t$ to camera motion $\mathbf{v}_{c_t}$ via an interaction matrix derived from the camera relative spatial velocity equation [25]; and ii) mapping $\mathbf{v}_{c_t}$ to robot motion $\mathbf{a_t}$ via a calibration model as in VS or a trial-error based online estimation as shown in Uncalibrated Visual Servoing (UVS [43]). Here we discuss feature-based visual servoing which our VGS learning directly links to.

VGS-IL removes the need for robust feature trackers while keeping the geometric error output that can be linked with a visual servoing controller. Compared to traditional approaches that hand select features to encode a task concept, VGS-IL directly learns the feature selection using a data driven approach. Instead of tracking each geometric feature and then associating them, VGS-IL directly extracts their associations in an adaptive manner which has been shown to be more robust [13].

It is worth noting that visual servoing control is sensitive to modeling errors [38]. Combining the 3D geometric vision aspect from VS to learn more robust controllers via Rein-

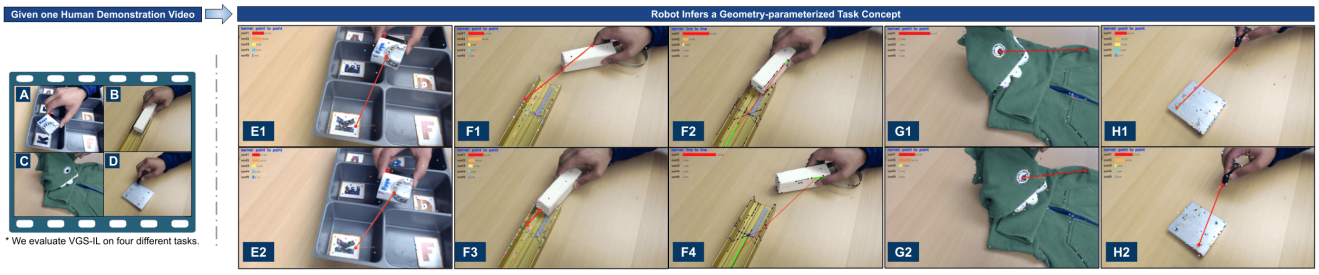

Fig. 5: Left: Four tasks designed in evaluation: *Sorting*, *Insertion*, *Folding* and *Screw*. **Right**: Qualitative evaluation results of VGS-IL in the four tasks. We select two frames for each task. The *Insertion* task includes two columns representing point-to-point and line-to-line kernel respectively. For a fair test, we changed the background and target pose in each task. Red line indicates selected feature association with highest confidence. Experiments show VGS-IL succeeds to learn a consistent geometry-parameterized task concept from human demonstrator in all the four tasks. Quantitative results are displayed in Table 1-2 below .

Task	*Sorting*		*Insertion: point-to-point*		*Insertion: line-to-line*		*Folding*		*Screw*	
Metrics	Acc	conAcc	Acc	conAcc	Acc	conAcc	Acc	conAcc	Acc	conAcc
Baseline1	100.0%	1.00	100.0%	1.00	100.0%	1.00	10.0%	n/a	8.2%	n/a
Baseline2	100.0%	0.03	100.0%	0.02	81.2%	-0.06	80.0%	0.10	33.0%	0.08
VGS-IL	100.0%	0.98	100.0%	0.85	93.0%	0.91	84.0%	0.98	49.0%	0.92

TABLE I: Quantitative evaluation of VGS-IL in the four tasks. All tests are based on changed background and randomly placed target. Results show VGS-IL performs better in learning a consistent geometry-parameterized task concept.

Settings	*Random Target*		*Change Camera*		*Object Occlusion*		*Object Outside FOV*		*Change Illumination*	
Metrics	Acc	conAcc	Acc	conAcc	Acc	conAcc	Acc	conAcc	Acc	conAcc
Baseline1	100%	1.00	0.0%	n/a	0.0%	n/a	0.0%	n/a	0.0%	n/a
Baseline2	99.1%	-0.03	96.7%	-0.10	92.7%	-0.05	81.2%	-0.03	0.0%	n/a
VGS-IL	100.0%	0.55	95.0%	0.61	97.3%	0.10	79.8%	0.19	19.2%	0.42

TABLE II: Evaluation results of VGS-IL on the robot imitator under different environmental settings (shown in Fig. 6). We keep testing on the real robot in the *Sorting* task, while exploring more variance settings. Results show VGS-IL performs the best under all conditions.

forcement Learning has the potential to derive both efficient and robust controllers.

IV. Experiments

Through experimental evaluation we aim to determine: (i) whether VGS-IL can learn a correct and consistent geometry-parameterized task concept given one human demonstration; and (ii) whether VGS-IL can output high-quality error signals for accurate robot control. For analysis, we decompose the two goals into four evaluation steps: (1) Given one human demonstration video, will VGS-IL output a correct and consistent task concept; (2) how will VGS-IL generalize from human demonstrator to robot imitator under changed task and environmental settings; and (3) How does control error converge, and how is it affected at different network training time for VGS-IL.

Baselines: We hand designed two baselines to use in comparison. *Baseline1* is conventional visual servoing with a video-tracking of a redundant feature set. This involves human interaction to carefully hand select 10 pairs of geometric features used to represent a task and initialize multiple feature trackers for each camera. As long as one pair out of ten is able to track throughout the entire task process, *baseline1* succeeds. *Baseline2* is a method from our previous work [13] that relies on hand crafted geometric feature descriptors (SIFT [19] and LBD [21]) in training; however, it doesn't take into consideration representation consistency.

Metrics: We designed two evaluation metrics: (1) *Acc* to measure accuracy; and (2) *conAcc* to measure consistency. Specifically, given N video frames, $Acc = \frac{M \times 100}{N}\%$, where M is the number of frames with correct geometric task concept inference. Defining *conAcc* is more challenging since directly measuring the inference consistency involves complex statistical methods [44]. For simplicity, we measure the time-series control error output $\{\mathbf{e}_t\}$ (i.e. the inference outcome) and define $conAcc = Autocorr(\{\|\mathbf{e}_t\|\}, k)$, which is the autocorrelation measurement over time-series error norms with *shift=k*. We fix *k=2* in all experiments. Since *baseline1* is a collection of redundant pairs of trackers, measuring the *conAcc* is difficult. In this case we assume that *conAcc=1*, the maximum, if *baseline1* succeeds.

Tasks: To facilitate comparisons, we follow the same four tasks: Sorting, Insertion, Folding, and Screw tasks as defined in [13] (see Fig. 5 for details). *Sorting* represents a rich texture clue task that requires a point-to-point kernel; the *Insertion* task needs a combination of point-to-point and line-to-line kernels; the *Folding* task represents deformable object manipulation; and the *Screw* task has low image textures.

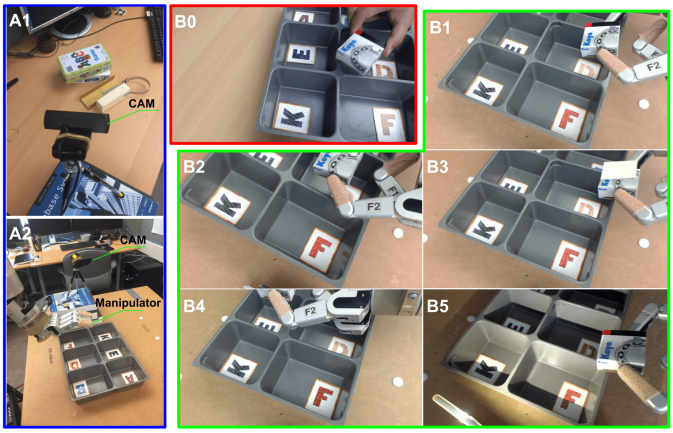

Fig. 6: A1: Human demonstration settings. **A2**: Robot imitation settings. **B0**: Human demonstration video used to train VGS-IRL. **B1-5**: Evaluation on robot under five different environmental settings. B1) *random target*; B2) *change camera*; B3) *object occlusion*; B4) *object outside camera's FOV*; B5) *change illumination*.

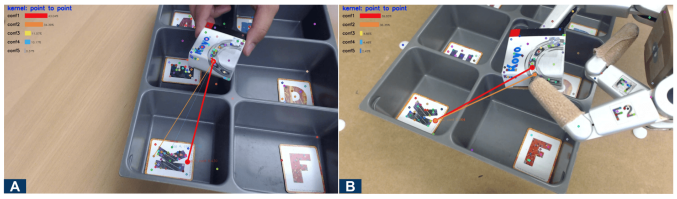

Fig. 7: Example of VGS-IL results in *Sorting* task. **A**: results in a human demo. **B**: results in a robot demo. Top five geometric feature associations are selected. Only the top one, as marked red color, is used in evaluation. Results show the same feature point association is selected regardless of human hand or robot hand under different backgrounds and target poses.

1) Evaluation on human demonstration videos: Our first step is to evaluate if VGS-IL learns a both correct and consistent geometric feature associations, given one human demonstration video. For a fair test, we changed both background and target pose in evaluation. Qualitative results are displayed in Fig. 5. Quantitative metric scores are shown in Table I. Results show VGS-IL succeeds to generalize the learned geometry-parameterized task concept in all the four tasks. Regarding selection consistency, VGS-IL performs the best compared to other two baselines.

2) Generalization under different environmental settings: Next we test if the learned task concept generalizes from human demonstrator to robot imitator. A WAM robot equipped with a Barret Hand is used to test the *Sorting* task (Fig. 7). Furthermore, we keep testing on the robot while exploring more variance settings (Fig. 6): (a) *random target*; (b) *move camera*: We test for real-world projective invariance by randomly translating and rotating the camera; (c) *object occlusion*; (d) *object outside FOV*: The object moves outside the camera's field of view and each method is required to automatically recover when the object is back in the image; and (e) *change illumination*: the lighting condition is changed by adding a spotlight light source. We pick the task *Sorting* to evaluate. Results are shown in Table II which indicate VGS-IL performs the best in all settings.

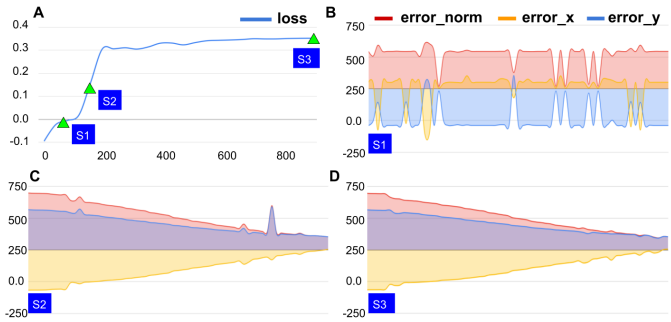

Fig. 8: Evaluation of the control error signals output from VGS-IL in the *Sorting* task. **A**: Training curve of VGS-IL with three different stages picked for evaluation. **B, C, D**: Control error signals output from VGS-IL trained in stage *S1, S2, S3*. Results clearly show that VGS-IL outputs a 'good' control error signal. Moreover, the optimization process is indeed optimizing the quality of control error signals.

3) Evaluation of the 'good' control error signal output: We test how 'good' or 'bad' the control error signals output from VGS-IL are. To do this, we had the robot perform the *Sorting* task via teleoperation, then ran VGS-IL on the resulting task video and measured the corresponding time-series error signals. Therefore, if VGS-IL was capable of outputting 'good' control signals, the results of this video should also be good. To make our evaluation more interesting, we wanted to see how control error signals are improved along with the optimization process of VGS-IL. Fig. 8 shows the results in three different training stages.

V. Conclusion

We present a geometric perspective on visual imitation learning. Specifically, we propose VGS-IL, visual geometric skill imitation learning, to learn a geometry-parameterized task concept. VGS-IL infers globally consistent geometric feature association rules from human demonstration video. The learned task concept outputs control error signals that can be directly linked to geometric vision based controllers, thus providing an efficient way to map learned high-level task concepts to low level robot actions. Experimental evaluations show that our method generalizes well from human demonstrator to robot imitator under various environmental settings.

In practice, VGS-IL needs large GPU computation resource due to its optimization over the whole combinatorial feature association candidates. A potential solution is to utilize high dimensional Bayesian Optimization methods [45] to directly estimate geometry representation and association parameters from the observation space. Moreover, although we demonstrated applying VGS-IL in tasks by combining different VGS kernels, it is worth further exploring how to sequentially link VGS kernels to program more complex tasks.

References

[1] 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.

[2] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," *Twenty-first international conference on Machine learning - ICML '04*, p. 1, 2004.

[3] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *International conference on machine learning*, 2016, pp. 49–58.

[4] J. Ho and S. Ermon, "Generative adversarial imitation learning," pp. 4565–4573, 2016.

[5] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," *arXiv preprint arXiv:1709.04905*, 2017.

[6] 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.

[7] C. J. Burke, P. N. Tobler, M. Baddeley, and W. Schultz, "Neural mechanisms of observational learning," *Proceedings of the National Academy of Sciences*, vol. 107, no. 32, pp. 14 431–14 436, 2010.

[8] D. Pathak, P. Mahmoudieh, G. Luo, P. Agrawal, D. Chen, Y. Shentu, E. Shelhamer, J. Malik, A. A. Efros, and T. Darrell, "Zero-shot visual imitation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2050–2053.

[9] M. Dehghan, Z. Zhang, M. Siam, J. Jin, L. Petrich, and M. Jagersand, "Online object and task learning via human robot interaction," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2132–2138.

[10] J. Jin, L. Petrich, M. Dehghan, Z. Zhang, and M. Jagersand, "Robot eye-hand coordination learning by watching human demonstrations: a task function approximation approach," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6624–6630.

[11] P. Sharma, D. Pathak, and A. Gupta, "Third-person visual imitation learning via decoupled hierarchical controller," in *Advances in Neural Information Processing Systems*, 2019, pp. 2593–2603.

[12] J. Jin, M. Dehghan, L. Petrich, S. W. Lu, and M. Jagersand, "Evaluation of state representation methods in robot hand-eye coordination learning from demonstration," *arXiv preprint arXiv:1903.00634*, 2019.

[13] J. Jin, L. Petrich, Z. Zhang, M. Dehghan, and M. Jagersand, "Visual geometric skill inference by watching human demonstration," *arXiv preprint arXiv:1911.04418*, 2019.

[14] Z. Dodds, G. D. Hager, A. S. Morse, and J. P. Hespanha, "Task specification and monitoring for uncalibrated hand/eye coordination," in *Proceedings 1999 IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1999, pp. 1607–1613.

[15] Z. Dodds, A. S. Morse, and N. Haven, "Task Specification and Monitoring for Uncali brat ed Hand / E ye Coordination *," no. May, 1999.

[16] Z. Dodds, M. Jägersand, and G. Hager, "A Hierarchical Architecture for Vision-Based Robotic Manipulation Tasks," *First Int. Conf. on Computer Vision Systems*, vol. 542, pp. 312–330, 1999.

[17] G. D. Hager and Z. Dodds, "On specifying and performing visual tasks with qualitative object models," *Proceedings-IEEE International Conference on Robotics and Automation*, vol. 1, no. April, pp. 636–643, 2000.

[18] S. R. Ahmadzadeh, A. Paikan, F. Mastrogiovanni, L. Natale, P. Kormushev, and D. G. Caldwell, "Learning symbolic representations of actions from human demonstrations," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3801–3808.

[19] D. G. Lowe *et al.*, "Object recognition from local scale-invariant features." in *iccv*, vol. 99, no. 2, 1999, pp. 1150–1157.

[20] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf." in *ICCV*, vol. 11, no. 1. Citeseer, 2011, p. 2.

[21] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.

[22] P. I. Corke, "High-performance visual closed-loop robot control," Ph.D. dissertation, 1994.

[23] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," *arXiv preprint arXiv:1806.08756*, 2018.

[24] P. Florence, L. Manuelli, and R. Tedrake, "Self-supervised correspondence in visuomotor policy learning," *IEEE Robotics and Automation Letters*, 2019.

[25] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[26] K. Ikeuchi and T. Suehiro, "Toward an assembly plan from observation. i. task recognition with polyhedral objects," *IEEE transactions on robotics and automation*, vol. 10, no. 3, pp. 368–385, 1994.

[27] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *IEEE transactions on robotics and automation*, vol. 10, no. 6, pp. 799–822, 1994.

[28] C. Xiong, N. Shukla, W. Xiong, and S.-C. Zhu, "Robot learning with a spatial, temporal, and causal and-or graph," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2144–2151.

[29] Y. Yang, Y. Li, C. Fermuller, and Y. Aloimonos, "Robot learning manipulation action plans by" watching" unconstrained videos from the world wide web," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[30] M. Sieb, Z. Xian, A. Huang, O. Kroemer, and K. Fragkiadaki, "Graph-structured visual imitation," *arXiv preprint arXiv:1907.05518*, 2019.

[31] Z. Qin, K. Fang, Y. Zhu, L. Fei-Fei, and S. Savarese, "Keto: Learning keypoint representations for tool manipulation," *arXiv preprint arXiv:1910.11977*, 2019.

[32] M. Gridseth, O. Ramirez, C. P. Quintero, and M. Jagersand, "Vita: Visual task specification interface for manipulation with uncalibrated visual servoing," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3434–3440.

[33] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1263–1272.

[34] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[35] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 109–116, 2004.

[36] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[37] G. J. Agin, *Servoing with visual feedback*. SRI International, 1977.

[38] F. Chaumette, "Geometric and photometric vision-based robot control: modeling approach," 2018.

[39] S. Levine, C. Finn, T. Darrell, and P. Abbeel, *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[40] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[41] Q. Bateux, "Going further with direct visual servoing," Ph.D. dissertation, Rennes 1, 2018.

[42] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke, "Training deep neural networks for visual servoing," in *ICRA 2018-IEEE International Conference on Robotics and Automation*, 2018, pp. 1–8.

[43] M. Jagersand, O. Fuentes, and R. Nelson, "Experimental evaluation of uncalibrated visual servoing for precision manipulation," *Proceedings of International Conference on Robotics and Automation*, vol. 4, no. April, pp. 2874–2880, 1997.

[44] T. Tarpey and B. Flury, "Self-consistency: A fundamental concept in statistics," *Statistical Science*, no. 3, pp. 229–243, 1996.

[45] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.