Act2Im: Action Conditioned Image Synthesis for Basketball

Anonymous ICCV submission

Paper ID 6071

Abstract

We introduce a novel action conditioned image synthesis task and a method to solve it in the context of a basketball activity. Given a target action category, which encodes an egocentric motion trajectory of a current player, our model synthesizes a visual signal of an opposing player that would possibly cause the current player to perform that action.

Our action conditioned image synthesis model consists of 1) a variational autoencoder (VAE), which generates masks of an opposing player, and 2) an ensemble of discriminative Action CNNs, which predict the next action of a current player. Initially, we train these two components of our system separately. Afterwards, we attach each Action CNN to the last layer of the VAE, and freeze the parameters of all the networks. During inference, given a target action category, we maximize its prediction probability at each Action CNN by backpropagating the gradients to the latent code of the VAE. Doing so iteratively, forces VAE to synthesize images that have high probability of a target action category. We show that our model generates realistic images that are associated with specific action categories, and it outperforms standard baselines by a large margin.

1. Introduction

Our behavior is inherently connected to the ability of perceiving visual information around us: what we see affects what we are going to do, and conversely what we do affects what we see. Consider a basketball player whose goal is to outmaneuver his/her defender and score a basket (see Figure 1 for an illustrative example). To accomplish this goal, he/she needs to carefully examine the stance of a defender, the position of a defender's feet, a defender's torso orientation, and many other factors about a defender. All this information is delivered via a player's visual perception system, which then allows a player to decide what to do next.

Unfortunately, understanding how humans leverage visual information in their decision making process is often challenging. For instance, in the context of basketball, most sub-second level decisions are made subconsciously by a player. Being able to explain why a skilled player chose one action over the other could be highly beneficial for developing tools that improve players' decision making ability.

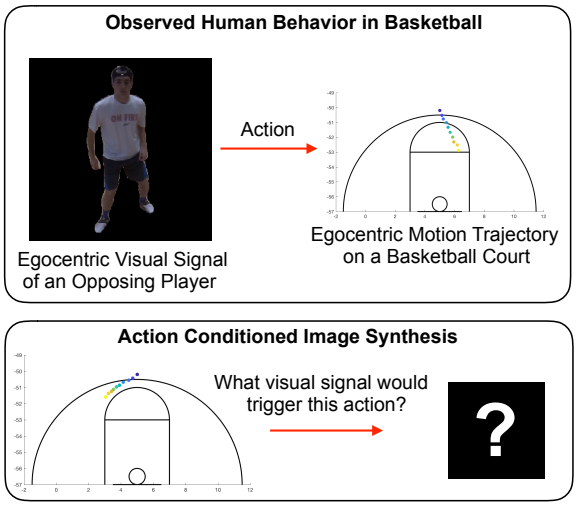

Figure 1: In a one-on-one basketball game, a player needs to outmaneuver his/her defender and score. Doing so requires assessing the stance of a defender, a defender's torso orientation, and many other factors. A player uses these visual cues about his/her defender to decide what action to perform next (i.e. how to move). This can be formulated as a problem of mapping a visual signal to an action. In this work, we aim to solve an inverse of this problem, which we refer to as an action conditioned image synthesis. Given a target action, we synthesize an image of an opposing player, which would likely trigger the player to perform that action.

Consider an augmented reality device in a form of glasses that could project an artificial image of an opposing player on its lenses. Suppose that such a device could synthesize and project a visual stimulus, which would cause the player who is wearing the glasses to perform a certain action. Why would this be useful? The players would benefit from practicing good decision making skills in a realistic, yet controlled environment with possibilities of simulating many diverse scenarios and getting real-time feedback after each of such scenarios.

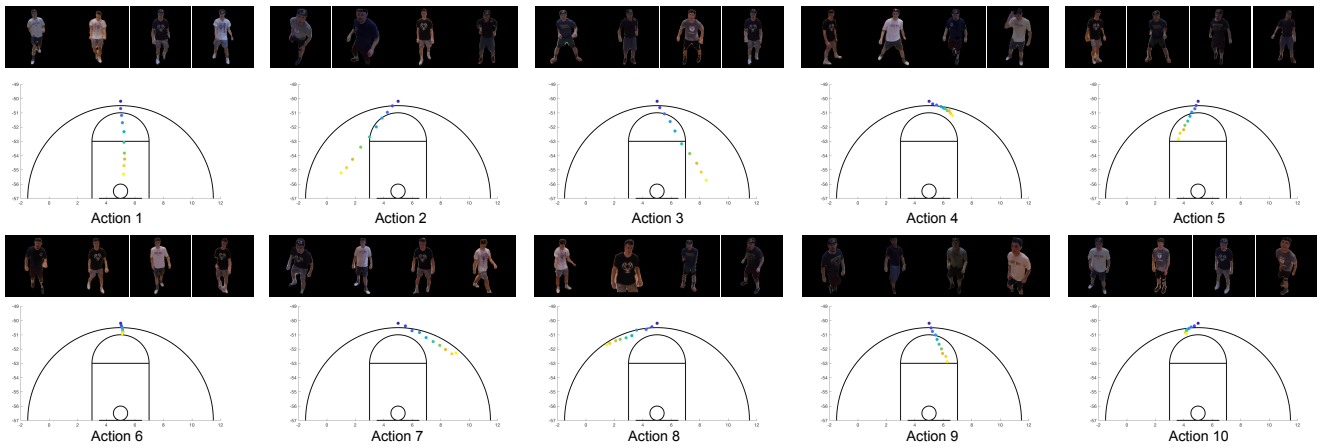

Figure 2: A figure illustrating all 10 of our discretized action categories, which are obtained via a K-means clustering. Each action category is visualized as an egocentric motion trajectory projected on a 2D ground plane of a basketball court. With each action category, we also include 4 images of an opposing player, which triggered those actions.

In order to build such a device we would first need a model that generates action triggering visual stimuli. We refer to this problem as action conditioned image synthesis. Given a target action in the form of an egocentric basketball motion sequence, we aim to generate an image of the opposing player that would likely trigger the current player to perform such an action (see Figure 1). We do so via our proposed action conditioned image synthesis model, which is composed of a variational autoencoder (VAE) [15] and an ensemble of discriminative convolutional neural networks (CNNs). During training, the VAE part of the system is trained to generate person-part masks of the opposing players. In parallel, we also train an ensemble of discriminative Action CNNs that take a person-part mask of an opposing player as input, and predict a discretized action category that encodes a player's future action (i.e. a discretized egomotion trajectory). During inference, each discriminative Action CNN is attached to the last layer of our trained VAE, and the parameters of all networks are frozen. Given a target action class, we compute the gradients that maximize the prediction of that action category, and backpropagate them all the way to the noise code, which is used by the VAE to generate the person-part mask of an opposing player. We use the computed gradients to iteratively adjust the noise code until the VAE generated mask produces maximum target action probabilitiy at each Action CNN.

In our experimental section, we show that given a target action class, our model successfully generates visual signals associated with that action category. Furthermore, we demonstrate that the gradients computed during action conditioned image synthesis procedure can be used to localize, which parts of an image are used most heavily to generate images for each action category. Finally, we show that our synthesized RGB images are quite similar to the ones that a player would see in a real basketball game.

2. Related Work

Computer Vision for Behavior Analysis. Developing models that analyze human behavior has been a fundamental problem in both computer vision and robotics. Recent work in [23, 31] leverages videos of humans performing daily tasks to teach robots the same behavior. The work in [40] develops a method that reminds humans of actions that they forgot to perform. The methods in [29, 24] develop techniques that predict walking trajectories for each person in the scene. Additionally, the works in [12, 25, 3, 2] leverage visual data for predicting pedestrian behavior.

Egocentric Vision. Many egocentric methods have recently been used to analyze human behavior. Most of these methods focused on active object detection [20, 8, 32, 10, 6] or activity recognition [16, 36, 35, 30, 21, 26, 9]. In addition to this work, the authors in [38] propose a model for detecting the camera wearer's engagement, while the work in [33] applies an inverse reinforcement learning technique to infer the goals of the camera wearer during daily tasks. Furthermore, the work in [28] introduces a model that generates walking trajectories from first-person RGBD data.

Behavior Analysis in Sports. In the past, there have been many attempts to use computer vision techniques for behavior understanding in sports. The method in [19], proposes a Markov-decision based technique to predict future trajectories of wide receivers in football. The work in [18] learns to predict motion trajectories of soccer players whereas the method in [42] leverages tracking data [1] and hierarchical CNNs for players' motion trajectory estimation in basketball. Finally, the work in [5] introduces a model to assess basketball skill from first-person videos, while the authors in [37, 4] leverage egocentric cameras to predict future motion trajectories of basketball players.

Our Contribution. In comparison to this prior work, we

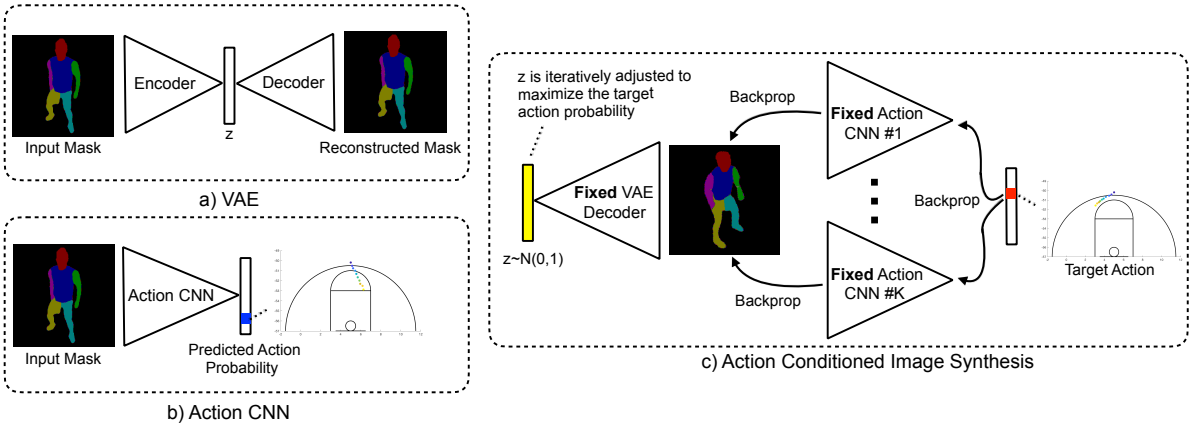

Figure 3: The figure illustrating each component of our action conditioned image synthesis model: **a)** a variational autoencoder (VAE), which is trained to generate person-part masks of an opposing player from a latent code z; **b)** Action CNN, which takes a person-part mask of an opposing player as its input, and predicts a current player's future action in a form of discretized egomotion trajectory projected on a 2D ground plane of a basketball court; **c)** a unified action conditioned image synthesis architecture obtained by attaching an ensemble of trained Action CNNs to the last layer of our trained VAE network, and freezing all the parameters. Given a target action category, the new person-part mask is synthesized by iteratively updating the latent code z such that the action probability in each Action CNN is maximized. Since our entire architecture is fully differentiable, the optimization problem can be solved using a standard backpropagation algorithm.

propose a novel action conditioned image synthesis task, and a new model for solving this task in the context of a basketball activity. We show that our model can synthesize images associated with particular action classes, and that it significantly outperforms standard baselines for this task.

3. Preliminaries

In this section, we discuss the reasons for using a one-on-one basketball setting for our action conditioned image synthesis task, and provide more details on the dataset that we use. Furthermore, we explain how we represent action space, and how we pre-process our input images.

3.1. Why Basketball?

Most prior computer vision research on human behavior analysis is done by studying daily activities such as cooking [20, 8, 32, 10, 6] or walking [28, 12, 25, 3, 2, 29, 24]. While these daily activities are well suited for tasks such as activity recognition, gaze prediction or motion trajectory estimation, they are not ideal for action conditioned image synthesis. We identify several key characteristics why a one-on-one basketball game is well suited for our task.

A one-on-one basketball game is advantageous because each player has the same goal (i.e. score a basket), but he/she can achieve this goal in numerous different ways (i.e. by using different motion trajectories to outmaneuver his/her defender). In contrast most daily activities such as cooking [7] or walking [28] are unscripted, which means that most people have different goals while they are performing those activities. We note that an action conditioned image synthesis task requires understanding a person's intention (i.e. how a person will use visual information to decide what action to perform), and thus, these daily activities are poorly suited for this task.

Furthermore, we believe that an adversarial component in a one-on-one basketball game (i.e. a competition against another player) is critical for our action conditioned image synthesis task. Consider a cooking activity and a particular action of "picking up a tomato". In this case, the tomato cannot move or change its appearance such that a person would perform a different action as a result of those changes. Thus, a person's actions in this case, are most heavily influenced by a person's intent, which is often unknown (due to the unscripted nature of these tasks). In contrast, in a one-on-one basketball game, every change in a defender's body can significantly alter the decision where a player will move next, which is perfect for our task.

Finally, we point out that using a one-on-one basketball setting is beneficial because action labels (i.e. motion trajectories) can be obtained without any manual annotations using Structure from Motion as in [37, 4]. In contrast, activities such as cooking require manual labeling of each action, which can be costly and time consuming [7].

3.2. Representing Action Space

For our experiments, we use an Egocentric One-on-One Basketball dataset from [4], which consists of 988 sequences from one-on-one basketball games. To select image-action pairs, we pick the first image from each sequence (before the player performs any action), and then

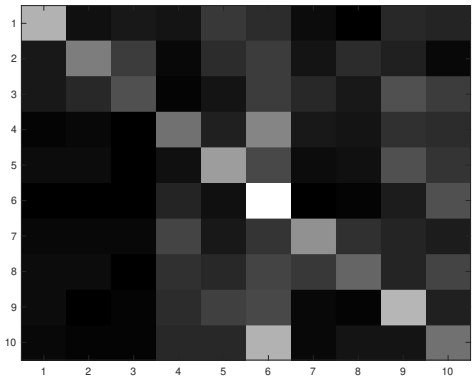

Figure 4: A figure illustrating the confusion matrix for the future action prediction task.

retrieve that player's motion trajectory for the next 2 seconds [4]. Through our experiments, we observed that predicting a concatenated (x, y) vector representation with limited amounts of training data is challenging. It is also difficult to interpret such representation since L2 or Hausdorff distance errors are not as intuitive as a classification error metric. Thus, to address these challenges, we discretize the entire action space into 10 action categories. To do this, we run a K-means clustering algorithm on the original (x, y) motion trajectories with 10 clusters. This allows us to represent each motion trajectory as one of the action clusters. We found that using such a discretized action representation makes it easier to train a classifier, and also allows us to interpret the experimental results easier. In Figure 2, we visualize these actions, as motion trajectories projected on a $2D$ basketball court in a top-view format. For each action category, we also include 4 images of an opposing player that triggered those actions.

3.3. Extracting Person-Part Masks

Instead of using raw RGB images of an opposing player as our input, we extract person-part masks that segment six body parts: head, torso, left arm, right arm, left leg, right leg [22]. We then use these person-part masks as input to the VAE, which learns to generate such masks from a latent code. We also use these masks as inputs to our action CNN, which predicts a player's future action from a given mask.

Operating on such person-part masks reduces the variance of factors such as players' clothing, players' appearance, etc. Considering that we have a limited amount of training data, this step allows us to prevent overfitting.

4. Model

We aim to synthesize images that are associated with a particular action of a basketball player. Our model has two main components: (1) a variational autoencoder (VAE) [15] that generates person-part masks of an opposing player, and (2) an ensemble of discriminative Action CNNs that predict the future action of a player from a given person-part mask. During training, the VAE and the discriminative Action CNNs are trained separately. Afterwards, during the image synthesis procedure, we attach each Action CNN to the last layer of the VAE, and freeze all the parameters (in both VAE and Action CNNs). Finally, given a target action, we maximize its predicted probability at each Action CNN by iteratively adjusting random noise code z, which is used as input to our trained VAE.

We illustrate each component of our proposed architecture in Figure 3, and describe all the details below.

4.1. Variational Autoencoder

For the VAE component of our model, we use a standard encoder-decoder network. Initially, the encoder f_E takes a person-part mask $x \in \mathbb{R}^{256\times256}$, and encodes it into a 32 dimensional latent representation $z \in \mathbb{R}^{1\times32}$. Then, the decoder f_D uses this latent space representation z to reconstruct the original person-part mask as $\hat{x} \in \mathbb{R}^{256\times256}$.

As is standard, we assume that the marginal distribution of the latent space z is a Gaussian with zero mean and identity covariance, i.e., $p(z) = \mathcal{N}(0, I)$. Under these assumption, the VAE maximizes the following quantity:

$$\sum_i E_{z\sim f_E}[\log f_D(x_i|z)] - D_{KL}(f_E(z|x_i)||p(z)) \quad (1)$$

The first term in the equation denotes expectation over distribution f_E, which is intuitively used to measure the accuracy of the decoder f_D for the distribution produced by the encoder f_E. The second term denotes Kullback-Leibler (KL) divergence, which penalizes f_E if it deviates too much from the desired distribution $p(z)$. In other words, the second term of the equation forces the latent space distribution z to follow a Gaussian distribution $\mathcal{N}(0, I)$. As a result, during inference we can generate a novel person-part mask $\hat{x}_i$ by sampling $z_i \sim \mathcal{N}(0, I)$ and feeding it through the decoder $f_D(z_i)$ thus, effectively eliminating the encoder f_E. We refer the reader to the work in [15] for more details on variational autoencoders.

Inspired by the success of image-to-image translation networks [13], we model our encoder, and decoder as a convolutional neural network (CNN) composed of 6 convolutional layers, and 6 deconvolutional layers. Furthermore, as is done in [17], we follow a generalized Bernoulli distribution to model $f_D(x_i|z)$.

4.2. Ensemble of Action CNNs

Given a person-part mask $x \in \mathbb{R}^{256\times256}$, the action CNN ϕ predicts a discrete action category $\phi(x) \in \mathbb{R}^{1\times M}$ that encodes one of M plausible future egocentric motion trajectories of a player. We implement ϕ using a popular ResNet-101 [11] architecture. The network is trained to predict

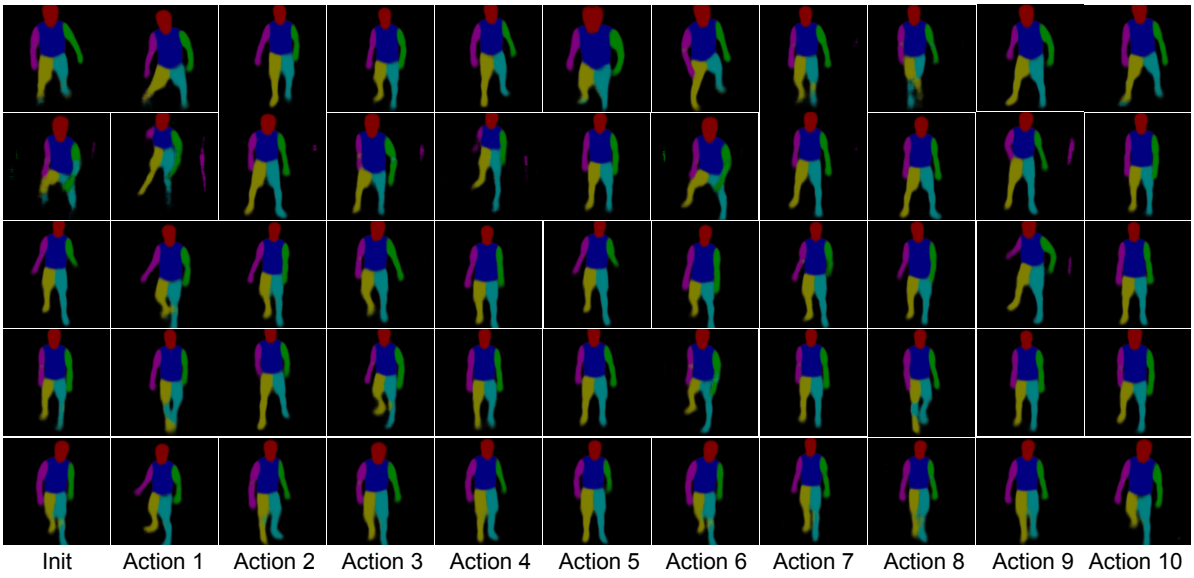

Figure 5: A figure illustrating the person-part masks generated via our action conditioned image synthesis procedure. Column 1 depicts masks that are generated during the first iteration from the initial latent code z. Other columns illustrate how the person-part masks for each action category diverge by the end of our optimization. These results suggest that our model generates substantially different outputs for each action category even when starting from the same latent code z.

ground truth action categories in a supervised fashion using a standard softmax loss function.

We aim to use our discriminative Action CNN ϕ as a supervisory signal for an action conditioned image synthesis procedure. Since a single Action CNN can be noisy, we instead train K independent Action CNNs ϕ^k where $k = 1...K$. Each of these Action CNNs are then used during our action conditioned image synthesis procedure, the details of which we describe below.

4.3. Action Conditioned Image Synthesis

We are now ready to put all the pieces of our model together and show how to synthesize images conditioned on a player's actions. Consider the two problems that we discussed previously: (1) action-agnostic person-part mask synthesis, and (2) a player's future action prediction. Our key insight is that by considering these two problems jointly, we can produce a model that generates images associated with a particular future action of a player.

To do so, we first attach each Action CNN to the last layer of our trained VAE (see Figure 3), and freeze all the parameters of such a joint model. Then, given an action $a \in \mathbb{R}^{1\times M}$, represented as a one-hot vector ($a_m = 1$, and $a_{m'} = 0$ for $m' \neq m$) we aim to find a latent code $z \in \mathbb{R}^{1\times 32}$ such that $\phi_m(f_D(z)) \in \mathbb{R}^{1\times M}$ is maximized, We formulate an action conditioned synthesis problem as the following maximization problem:

$$z^* = \arg\max_z \sum_{k=1}^{K} -\log\left(\phi_m^k(f_D(z))\right) \qquad (2)$$

where ϕ^k is each of K discriminative Action CNNs, m depicts an index of a target action category, and f_D is a VAE decoder that takes a latent code z and generates a person-part mask $\hat{x} \in \mathbb{R}^{256\times 256}$. Note that because each step of our unified architecture is differentiable, the following optimization problem can be easily solved using a standard backpropagation algorithm. During the maximization, all the parameters are frozen, and we only adjust the latent code z, which is initialized to a random sample from $\mathcal{N}(0, I)$. Once we discover the z^* that maximizes the objective above, we can generate a person-part mask by feeding it through the VAE decoder as $f_D(z^*)$. Finally, by feeding our synthesized person-part masks through an image-to-image translation network [17], we obtain RGB representation of our generated outputs.

4.4. Implementation Details

Our VAE was trained on 256×256 resolution images for 50 epochs using a batch size of 30, a learning rate of 0.0002 and Adam optimizer [14]. The encoder of the VAE consisted of six 4×4 convolutional layers with $64, 128, 256, 512, 512, 512$ output channels respectively, each layer followed by a leaky ReLU, and batch normalization. The decoder of the VAE consisted of six 4×4 deconvolution layers with $512, 512, 512, 256, 128, 64$ output channels respectively, each layer followed by a ReLU and batch normalization. The final output of the VAE was a $6\times256\times256$ person-part mask encoding six distinct human body parts: head, torso, both arms, and both legs.

Method	Accuracy ↑	Mean F-score ↑
Random	0.124	0.090
Action CNN	**0.377**	**0.340**

Table 1: Future action prediction results according to the standard accuracy and the mean per-class F-score metrics. Our Action CNN significantly outperforms a random baseline according to both metrics, suggesting that it learned to associate visual inputs to a player's future actions reasonably well.

Each Action CNN was based on a ResNet-101 [11] design and was trained for $40K$ iterations, with a learning rate of 0.00025, 0.9 momentum, a weight decay of 0.0005, and 10 samples per batch using SGD optimizer. We trained 7 independent Action CNNs, and used 6 of them for our action conditioned synthesis procedure, and one of them for evaluation purposes. During action conditioned image synthesis procedure, we adjusted the latent code z using backpropagation with a learning rate of 0.05 until the loss at each Action CNN became less than 0.3 or until the optimization exceeded 150 iterations. To generate RGB images from the synthesized person-to-part masks we adopted an image-to-image translation network from [17], which we trained and tested exactly as was done in the original paper.

5. Experimental Results

To the best of our knowledge, we are the first to tackle the problem of action conditioned image synthesis. Therefore, there are no previously established baselines to assess the performance of our model. As noted in [13], automatic evaluation of synthesized images is still an open and a challenging problem. In the context of our action conditioned image synthesis task, the evaluation becomes even more challenging because unlike for tasks such as image generation, or object detection, the solution to our problem is not obvious. Predicting what a player will do in the future is highly uncertain, which also makes it difficult to assess whether our synthesized images align well with a particular target action category. Finally, the evaluation problem is exacerbated by the fact that our task requires a domain specific (i.e. basketball) expertise, which takes years of training to obtain.

To address these shortcomings, we propose to evaluate action conditioned image synthesis task automatically using a held-out Action CNN that has not been previously used for image synthesis. In order for this evaluation scheme to be valid, we must first show that Action CNN learns to reliably associate input images to a player's future actions. Once we verify that our trained Action CNN learned reliable image-action associations, we can feed our generated images through a previously unseen Action CNN, and compute the loss of that image for a given action category.

We point out that the evaluation techniques relying on pre-trained CNNs have been widely used by prior methods [13, 34, 39, 41, 27]. We also note that to complement our quantitative results, we include a wide array of qualitative results.

Method	Test Loss ↓
Action Agnostic VAE	9.851
Action Specific VAE	6.347
Ours w/ Single Action CNN	4.028
Ours w/ Ensemble of Action CNNs	**1.296**

Table 2: Our action conditioned image synthesis results evaluated using a loss of a previously unseen Action CNN (the lower the better). As our first baseline, we include a standard VAE trained to generate person-part masks of opposing players in an action agnostic fashion. Our second baseline involves training a separate VAE for each action category. We also include two variations of our own model: one that uses a single Action CNN during an action conditioned image synthesis procedure, and a stronger model that uses an ensemble of 6 Action CNNs for the same purpose. We show that our strongest model outperforms these baselines by a large margin.

5.1. Quantitative Results

Future Action Prediction. We first want to verify that our trained Action CNN can reliably predict a player's future action from a given person-part mask. To maximize the amount of training data, we train our Action CNN using a leave-one-out cross validation with 8 splits. We also augment our dataset by horizontally mirroring input images and the corresponding ground truth motion trajectories. This leads to about 1,700 image-action pairs per split. We then evaluate our results according to two metrics: (1) total accuracy, and (2) mean per-class F-score, which is a more reliable metric to deal with class imbalance.

In Table 1, we report action prediction accuracy averaged over 8 splits. Our trained Action CNN achieves a 37.7% total accuracy and a 0.34 mean per-class F-score, significantly outperforming a random baseline, which yields 12.4% total accuracy and a 0.09 mean per-class F-score. In Figure 4, we also visualize the confusion matrix illustrating the performance for each of 10 action classes.

Considering the difficulty of this task, we think that these results are solid. We would also like to point out that our goal in this paper is not to build the most accurate action prediction model. We simply want to design an action prediction model that is good enough for the purposes of an action conditioned image synthesis task (and its evaluation).

Action Conditioned Image Synthesis. To evaluate action conditioned image synthesis task, we generate $1,000$ images for each cross validation split[1] (100 per each action

[1]We maintain the original cross validation splits from the action prediction task by making sure that the ensemble of Action CNNs used for

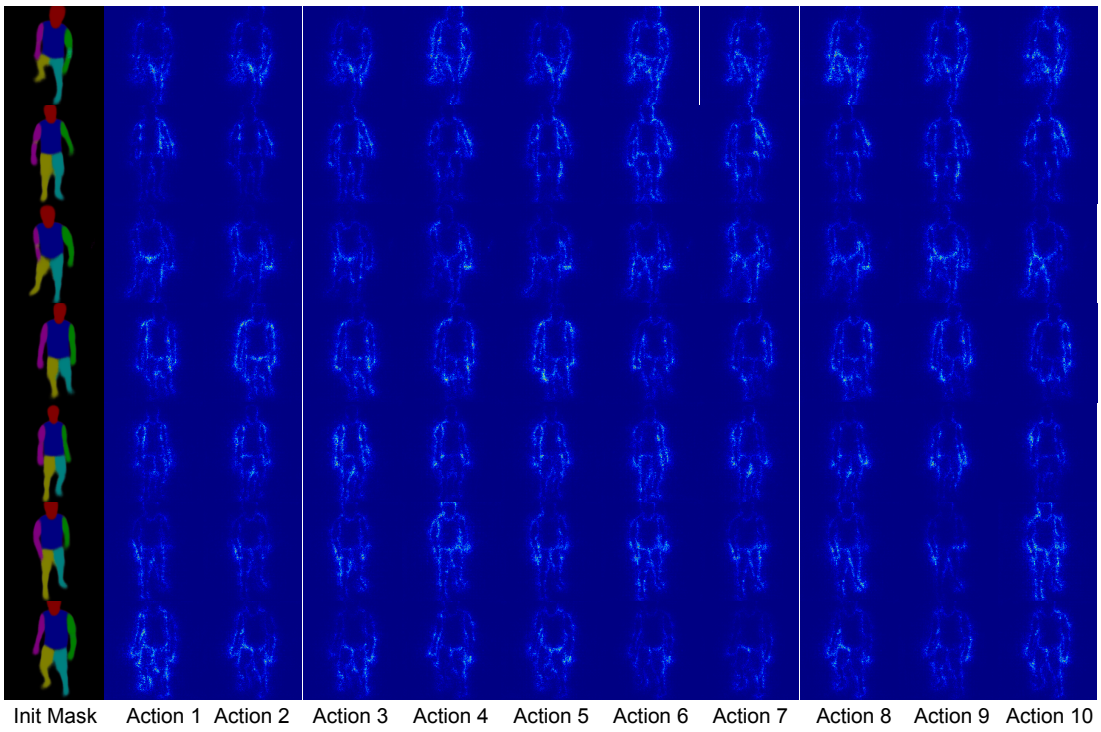

Figure 6: To understand what visual cues our model deems most important for each action category, we visualize the gradients that are used to modify the initial person-part masks during our action conditioned image synthesis procedure. Column 1 illustrates the person-part masks that are generated from the initial latent code z. Other columns illustrate the gradients that are used to adjust the initial mask for each action category. From these visualizations, we observe that different body parts are considered relevant for different action categories.

class). We then feed each synthesized image $\hat{x}_i(m)$ with an action category m, to a previously unseen Action CNN ϕ^{test}, and compute its loss as:

$$L_{test} = \frac{1}{N}\sum_{i}^{N} -\log\left(\phi_m^{test}(\hat{x}_i(m))\right) \quad (3)$$

We compute L_{test} for each cross validation split and then average these values across all splits. The final loss values are reported in Table 2, where we compare our model with a few other relevant baselines. As one of our baselines, we include a standard VAE. which learns to generate person-part masks that are action agnostic. As our second baseline, we include a model where we train a separate VAE for each action category (10 separate VAEs in this case). Such a baseline is very costly, but we still include it to demonstrate that our model outperforms it by a large margin. Finally, we include two variations of our model: 1) when only a single Action CNN is used for action conditioned image synthesis, and 2) when we use an ensemble of Action CNNs.

Based on these results, we observe that our best model yields a loss of 1.296, which is significantly lower than the loss values produced by two standard baselines (9.851 and 6.347). Furthermore, we notice that using an ensemble of Action CNNs during an action conditioned image synthesis procedure improves our performance from 4.028 to 1.296, which is a substantial boost. We used 6 Action CNNs– a maximum number that could fit in a 12GB GPU machine.

image synthesis have been trained on the appropriate split.

5.2. Qualitative Results

Visualizing Generated Person Masks. In Figure 5, we visualize person-part masks generated by our model for each action category. Each row of Figure 5, depicts masks generated from the same initial latent code z. The mask that is generated by the VAE during the first iteration is illustrated in Column 1 of Figure 5 under the name "Init". In further iterations, the masks for different action categories diverge as the gradients from Action CNNs are different for each action category. We observe that even if we start with the exact same latent code z, the model ends up generating substantially different outputs for each action category. This result indicates that our model can differentiate, which visual cues are more important for each action category.

In fact, to understand what visual cues are most important for each action category, we visualize the gradients that are used to update the masks for each action category during action conditioned image synthesis. We present these visualizations in Figure 6. Images in Column 1 of Figure 6

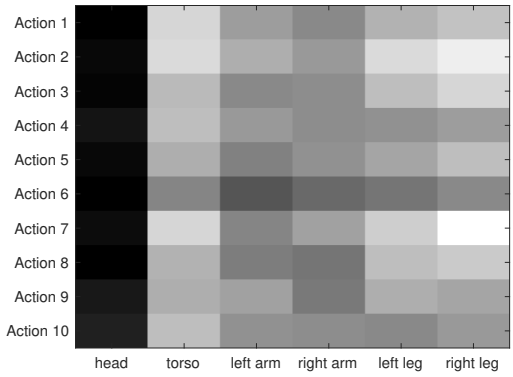

Figure 7: We average the gradients computed during an action conditioned image synthesis procedure across the regions corresponding to each body part. We do this for each action category across the entire set of our generated images. These results are then visualized as a 2D table where the white regions indicate large gradient values, and the dark regions depict small gradient values. From these visualizations, we observe that the legs, and the torso are much more informative than the head and the arms, which makes intuitive sense. Furthermore, we notice that gradients for action category 6 are lower, which also makes sense since this action defines a relatively stationary motion trajectory.

depict person-part masks that are synthesized from the initial latent code z. The remaining columns in the figure illustrate the gradients that are used to modify the initial mask for a particular action category (during a subsequent iteration). Based on these visualizations, we observe that for each action category, our model chooses to modify the areas corresponding to different body parts. For instance, the example in the top row shows that in this particular case, the gradients around torso are not very important for action categories $1, 2, 3$ and 5. However, for that same example, the gradient values around torso are pretty large for action categories 4 and 6.

To generalize these findings, we average the gradient values in the regions associated with each body part across the entire set of our synthesized images. We then visualize these values for each action category in Figure 7. The lighter color means higher gradient values, whereas the darker color depicts smaller gradient values. These results provide some interesting insights into what our model had learned. For instance, we can observe that gradients for action category 6 are generally smaller than for all the other action categories. This makes intuitive sense because action category 6 encodes a pretty stationary motion trajectory (See Figure 2), and thus, visual cues become less relevant in that case. Furthermore, we observe that in general, the most important body parts are legs and torso. This also makes intuitive sense because visual cues around the opposing player's head or arms are not very informative. In contrast, it makes sense that the model "fires" on pixels around torso, or legs because these body parts are the most important for determining how quickly a defender can move one way or the other.

Figure 8: A figure illustrating our synthesized RGB basketball images. First, we use an image-to-image translation network [17] to generate RGB images from our previously synthesized person-part masks. Afterwards, we project an RGB image of an opposing player on an empty egocentric basketball image of a basketball court with no players in it. Based on these results, we observe that unless we zoom-in, these images look quite similar to what a player might see in a real one-on-one basketball game.

Visualizing Generated RGB Images. We use an image-to-image translation network [17] to transform our synthesized person-part masks to the RGB images of an opposing player. We then project those images, on an empty egocentric basketball image of a basketball court without any players in it. We visualize this result in Figure 8. Based on these results, we observe that our generated RGB basketball images look pretty realistic unless we zoom in to look at various fine-grained details such as faces or clothing texture. We believe that this is a promising result allowing many potential basketball applications in the future.

6. Discussion

In this work, we introduced a novel action conditioned image synthesis task, and a model to solve it in a basketball setting. One limitation of our model is the assumption that given the same visual signal every player will act the same way. In our case, making this assumption is necessary due to a limited amount of training data. However, in our future work, we plan to address this limitation by collecting more data and exploring models that are personalized for each player. Furthermore, we note that due to a general design of our model, we plan to explore action conditioned image synthesis task in the context of other activities as well, not just a basketball game.

References

[1] https://www.stats.com. 2

[2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *CVPR*, 2016. 2, 3

[3] A. Alahi, V. Ramanathan, and F.-F. Li. Socially-aware large-scale crowd forecasting. In *CVPR*, pages 2211–2218. IEEE Computer Society, 2014. 2, 3

[4] G. Bertasius, A. Chan, and J. Shi. Egocentric basketball motion planning from a single first-person image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 3, 4

[5] G. Bertasius, H. S. Park, S. X. Yu, and J. Shi. Am i a baller? basketball performance assessment from first-person videos. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2017. 2

[6] G. Bertasius, H. S. Park, S. X. Yu, and J. Shi. First-person action-object detection with egonet. In *Proceedings of Robotics: Science and Systems*, July 2017. 2, 3

[7] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[8] D. Damen, T. Leelasawassuk, O. Haines, A. Calway, and W. Mayol-Cuevas. You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014. 2, 3

[9] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *ICCV*. 2

[10] A. Fathi, X. Ren, and J. M. Rehg. Learning to recognize objects in egocentric activities. In *CVPR*, pages 3281–3288. IEEE Computer Society, 2011. 2, 3

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, 2015. 4, 6

[12] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282–4286, May 1995. 2, 3

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arxiv*, 2016. 4, 6

[14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[15] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. 2, 4

[16] K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *CVPR*, 2011. 2

[17] C. Lassner, G. Pons-Moll, and P. V. Gehler. A generative model for people in clothing. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 4, 5, 6, 8

[18] H. Le, P. Carr, Y. Yue, and P. Lucey. Data-driven ghosting using deep imitation learning. In *MITSSAC*, 2017. 2

[19] N. Lee and K. M. Kitani. Predicting wide receiver trajectories in american football. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016. 2

[20] Y. J. Lee and K. Grauman. Predicting important objects for egocentric video summarization. *IJCV*, 2015. 2, 3

[21] Y. Li, Z. Ye, and J. M. Rehg. Delving into egocentric actions. In *CVPR*. 2

[22] G. Lin, A. Milan, C. Shen, and I. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 2017. 4

[23] Y. Liu, A. Gupta, P. Abbeel, and S. Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. *CoRR*, abs/1707.03374, 2017. 2

[24] W.-C. Ma, D.-A. Huang, N. Lee, and K. M. Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 3

[25] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *CVPR*, pages 935–942. IEEE Computer Society, 2009. 2, 3

[26] K. K. Minghuang Ma. Going deeper into first-person activity recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[27] A. Owens, P. Isola, J. H. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman. Visually indicated sounds. In *CVPR*, pages 2405–2413. IEEE Computer Society, 2016. 6

[28] H. S. Park, J.-J. Hwang, Y. Niu, and J. Shi. Egocentric future localization. In *CVPR*, 2016. 2, 3

[29] S. Pellegrini, A. Ess, K. Schindler, and L. J. V. Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, pages 261–268. IEEE Computer Society, 2009. 2, 3

[30] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, 2012. 2

[31] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. *CoRR*, abs/1707.02920, 2017. 2

[32] X. Ren and C. Gu. Figure-ground segmentation improves handled object recognition in egocentric video. In *CVPR*, 2010. 2, 3

[33] N. Rhinehart and K. M. Kitani. First-person activity forecasting with online inverse reinforcement learning. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[34] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc., 2016. 6

[35] S. Singh, C. Arora, and C. V. Jawahar. First person action recognition using deep learned descriptors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[36] B. Soran, A. Farhadi, and L. Shapiro. *Action Recognition in the Presence of One Egocentric and Multiple Static Cameras*. 2015. 2

[37] S. Su, J. P. Hong, J. Shi, and H. S. Park. Predicting behaviors of basketball players from first person videos. In *CVPR*, 2017. 2, 3

[38] Y.-C. Su and K. Grauman. Detecting engagement in egocentric video. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[39] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *ECCV*, 2016. 6

[40] C. Wu, J. Zhang, B. Selman, S. Savarese, and A. Saxena. Watch-bot: Unsupervised learning for reminding humans of forgotten actions. 2016. 2

[41] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *ECCV*, 2016. 6

[42] S. Zheng, Y. Yue, and J. Hobbs. Generating long-term trajectories using deep hierarchical networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1543–1551. Curran Associates, Inc., 2016. 2