Stochastic Actor-Executor-Critic for Image-to-Image Translation

Ziwei Luo¹*, Jing Hu¹*, Xin Wang²†, Siwei Lyu³, Bin Kong², Youbing Yin², Qi Song² and Xi Wu¹†
¹Chengdu University of Information Technology, China
²Keya Medical, Seattle, USA
³University at Buffalo, SUNY, USA
xi.wu@cuit.edu.cn, xinw@keyamedna.com

Abstract

Training a model-free deep reinforcement learning model to solve image-to-image translation is difficult since it involves high-dimensional continuous state and action spaces. In this paper, we draw inspiration from the recent success of the maximum entropy reinforcement learning framework designed for challenging continuous control problems to develop stochastic policies over high dimensional continuous spaces including image representation, generation, and control simultaneously. Central to this method is the Stochastic Actor-Executor-Critic (SAEC) which is an off-policy actor-critic model with an additional executor to generate realistic images. Specifically, the actor focuses on the high-level representation and control policy by a stochastic latent action, as well as explicitly directs the executor to generate low-level actions to manipulate the state. Experiments on several image-to-image translation tasks have demonstrated the effectiveness and robustness of the proposed SAEC when facing high-dimensional continuous space problems.

1 Introduction

Many computer vision problems, such as face inpainting, semantic segmentation, and realistic photo generated from sketch, can be defined as learning image-to-image translation (I2IT). Currently the most effective I2IT solution is based on a one-step framework that generates images in a single run of a deep learning (DL) model, such as VAE, U-Net, and conditional GANs. Directly learning I2IT with these DL models is challenging, due to the abundance of local minimums and poor generalization caused by overfitting. Although these problems could be potentially alleviated by using multi-scale models or a multi-stage pipelines, we are still left with models that have intrinsically high complexities, for which the optimal parameters (e.g. stage number and scale factor) have to be determined in a subjective and ad hoc manner.

To address these limitations of DL-based methods, we explore solving I2IT problems by leveraging the recent advances in deep reinforcement learning (DRL). The key idea is to decompose the monolithic learning process into small steps by a lighter-weight CNN, with the aim of progressively improving the quality of the model. Although recent works have successfully applied DRL to solve several visual tasks [Caicedo and Lazebnik, 2015], the action space in their tasks is usually discrete and can not be used for I2IT that requires continuous action spaces.

Currently, a promising direction for learning continuous actions is the maximum entropy reinforcement learning (MERL), which improves both exploration and robustness by maximizing a standard RL objective with an entropy term [Haarnoja et al., 2018]. Soft actor-critic (SAC) [Haarnoja et al., 2018] is an instance of MERL and has been applied to solve continuous action tasks. However, the main issue hindering the practical applicability of SAC on I2IT is its inability to handle high-dimensional states and actions effectively. Although recent work [Yarats et al., 2019] addresses this problem by combining SAC with a regularized autoencoder (RAE), RAE only provides an auxiliary loss for an end-to-end RL training and is incapable for I2IT tasks.

Besides, a glaring issue is that the reward signal is so sparse that leads to an unstable training, and higher dimensional states such as pixels worsen this problem [Yarats et al., 2019]. Thus, training an image-based RL model requires much more exploration and exploitation. One solution to stabilize training is to extract a lower dimensional visual representation with a separately pre-trained DNN model, and learn the value function and corresponding policy in the latent spaces [Nair et al., 2018]. However, this approach can not be trained from scratch, which could lead to inconsistent state representations with an optimal policy. Previous works like [Lee et al., 2020] have shown that it is beneficial to learn the image representations and continuous controls simultaneously.

In this paper, we propose a new DRL architecture, stochastic actor-executor critic (SAEC), to handle I2IT with very high dimensional continuous state and action spaces. SAEC is formed by three core deep neural networks: the actor, the critic, and the executor (see Fig. 1). High level actions are generated by the actor in a low dimensional latent space, and forwarded to the executor for image translation. The critic evaluates the latent actions to guide the policy update. These
three neural networks form two different branches in SAEC. More specifically, the actor and the critic form the RL branch as an actor-critic model, while the actor and executor form the DL branch similar to autoencoder, with an important difference that skip connections are added between the actor and executor. The RL branch is trained based on the MERL framework, while the DL branch is trained by minimizing a reconstruction objective.

Our contributions can be summarized as follows:

• A new DRL framework stochastic actor-executor critic (SAEC) is proposed to handle very high dimensional state and action spaces in I2IT task.
• Even using a sparse reward signal, our SAEC model can be stably trained from scratch on I2IT problems by combining DL-based supervision.
• Our SAEC framework is flexible to incorporate many advanced DL methods for various complex I2IT applications. And this framework enables the agent to simultaneously learn feature representation, control, and image generation in high-dimensional continuous spaces.

2 Background

2.1 Image-to-Image Translation

Image-to-image translation (I2IT) is often addressed by learning a generative process \( G \) that maps state \( x \) to target \( y \), \( G : x \rightarrow y \). The state should be consistent with the target and both of them are images, such as generating realistic photos from semantic segmentation labels [Isola et al., 2017], or synthesizing completed visually targets from images with missing regions [Pathak et al., 2016]. Autoencoder is leveraged in most research work to learn this process by minimizing the reconstruction error \( \mathcal{L}_{rec} \) between the predicted image \( \hat{y} \) and the target \( y \) (see Figure 1). In addition, the generative adversarial network (GAN) is also vigorously studied in I2IT to synthesis realistic images [Isola et al., 2017]. Subsequent works enhance I2IT performance by using a coarse-to-fine deep learning framework [Yu et al., 2018] that recursively uses the previous stage’s output as the input to the next stage (see the bottom of the Fig. 1(a)). In this way, I2IT task is transformed into a multi-stage, coarse-to-fine solution. Although iteration can be infinitely applied in theory, it is limited by the increasing model size and training instability.

2.2 Reinforcement Learning

Reinforcement learning (RL) is an infinite-horizon Markov decision process (MDP), defined by the tuple \( (\mathcal{X}, \mathcal{A}, P, r, \gamma) \). \( \mathcal{X} \) is a set of states, \( \mathcal{A} \) is action space, and \( P \) represents the state transition probability given \( x \in \mathcal{X} \) and \( a \), \( r \) is the reward emitted from each transition, and \( \gamma \in [0, 1] \) is the reward discount factor. The standard objective of RL is to maximize the expected sum of rewards. Maximum entropy RL (MERL) further adds an entropy term: \( \sum_{t=0}^{T} \mathbb{E}_{(x_t, a_t) \sim \pi} [r_t + \alpha \mathcal{H} (|x_t|)] \), where \( \alpha \) is a temperature parameter, \( \rho_x \) denotes the marginal trajectory distribution induced by policy \( \pi \). MERL model and has proven stable and powerful in low dimensional continuous action tasks, such as games and robotic controls [Haarnoja et al., 2018]. However, when facing complex visual problems such as I2IT, where observations and actions are high-dimensional, it remains a challenge for RL models [Lee et al., 2020].

3 Method

This work reformulates I2IT as a decision making problem. Instead of directly mapping the input to the target image in a single step, we propose to use a light-weight RL model that performs the translation progressively, where new details can be added as the translation progresses. The proposed model stochastic actor-executor-critic (SAEC) consists of three components: an actor \( (\pi_\theta) \), an executor \( (\eta_\omega) \), and a critic \( (Q_\phi) \). Specifically, the actor \( \pi_\theta \) and the executor \( \eta_\omega \) form a DL pathway that directs the agent learning with a DL-based objective. The same actor and the critic \( Q_\phi \) form a DRL pathway that works as an actor-critic model. In SAEC, the actor generates a latent action according to a stochastic policy, so as to capture the required image translation control. The critic evaluates the stochastic policy. The executor leverages the latent action and image details to perform image translation and the generated image is then applied to the environment as the new input image at next time.

3.1 I2IT as MDP

The key of our method is to formulate I2IT as a Markov decision process (MDP). As for state \( x_t \in \mathcal{X} \), it is defined task specific. For realistic photo generation or segmentation, the state \( x_t \) is the pair of the fixed source image \( I_f \) and the moving image \( I_m \) at time \( t \). A new state is generated by warping the moving image with predicted image action \( \hat{y} \): \( x_{t+1} = (I_f, \hat{y} \circ I_m) \). A face inpainting, the state is composed of the original image \( x \) and the synthesized missing part \( \hat{y} \). The next state is obtained by adding the new predicted image to the missing area: \( x_{t+1} = (\hat{y} \circ x_t) \).

There are two types of actions in our action space: latent action \( z_t \in \mathcal{Z} \) and image action \( y_t \in \mathcal{Y} \). These two actions are intrinsically different, in that \( \mathcal{Z} \) has a high level of abstraction and its distribution is unknown. \( z_t \) is sampled from the stochastic policy of the actor \( \pi_\theta \), \( z_t \sim \pi_\theta (x_t) \) while \( \hat{y} \) is created by the executor based on \( z_t \) and the input image. The reward function is flexible and can be any evaluation metric that measures the similarity between the predicted image \( \hat{y} \) and the target \( y \), such as the peak signal-to-noise ratio (PSNR) and the structural similarity index measure (SSIM).

The sequence of actions are implemented by the stochastic actor-executor-critic (SAEC) model. Specifically, at each time step \( t \), the actor samples a latent action \( z_t \) from state \( x_t \), then the executor generates image action \( \hat{y}_t \) from the latent action \( z_t \) and state \( x_t \). The change of the state will result in a reward \( r_t \), and the new state is fed back to the SAEC model to generate a new image action. Let \( T_t \) represents a single run from state \( x_t \), to image action \( \hat{y}_t \), via an actor and an executor: \( z_t \sim \pi_\theta (x_t), \hat{y}_t = \eta_\omega (x_t, z_t) \). The whole process of I2IT in MDP can be formulated as follows:

\[
T(x) = T_t \circ T_{t-1} \circ \cdots \circ T_0(x),
\]

where \( \circ \) is the function composition operation, \( T_t \) is the \( t \)th translation step that predicts image from current state \( x_t \).
Figure 1: The conventional DL frameworks and our SAEC framework. Top of (a): One-stage DL framework. Bottom of (a): Multi-stage DL framework, whose networks are usually different in each stage. (b) Our SAEC framework. x and y are input and target. ŷ is the predicted output. x_t and ŷ_t are temporary inputs and outputs generated in step (stage) t. In the SAEC framework, there are two actions: high-level latent-action z_t, which is sampled from policy π_φ: z_t ∼ π_φ(z_t|x_t), and low-level image-action ŷ_t which is generated from executor η_ψ conditioned by z_t: ŷ_t = η_ψ(ŷ_t|x_t, z_t). Both actions are high-dimensional continuous.

### 3.2 Policy with Skip Connections

As in an MERL model, the stochastic policy and maximum entropy in our SAEC improve the exploration for more diverse generation possibilities, which helps to prevent agent from producing a single type of plausible output during training (known as mode-collapse). In addition, one specific characteristic in the SAEC model is that skip connections are added from each down-sampling layer of the actor to the corresponding up-sampling layer of the executor, as shown in Fig. 1(b). In this way, a natural looking image is more likely to be reconstructed since the details of state x_t can be passed to the executor by skip connections. Besides, since both z_t and x_t can be used by the executor to generate image action, over-exploration of the action space can be avoided in SAEC, where the variance is limited by these passed detail information. Furthermore, the skip connections also facilitate back-propagation of the DL gradients to the actor. Our experiments show that if the skip connections are removed, the details of the reconstructed image may be lost because of down-sampling (max-pooling) and stochastic policies, and the training would be very unstable in such a situation. Combining skip connections and DL-based learning can avoid detail loss and thus stabilize training. (See experiments section for more details).

### 3.3 Stochastic Actor-Executor-Critic

To address I2IT, we propose to extend the maximum entropy RL (MERL) framework with an additional executor. Our proposed model has a similar structure with SAC [Haarnoja et al., 2018], but the latter cannot be applied to the I2IT tasks directly since the delayed and sparse rewards making it difficult to handle the inherent high-dimensional states and actions. To this end, we propose to train the actor and executor in a conventional DL-based manner, where the training data are collected from the agent-environmental interaction and change dynamically in experience replay. The objective of SAEC is to learn a policy π_φ and an executor η_ψ, by maximizing both the conditional likelihood and the expected sum of rewards with an entropy H.

#### Conditional Generative Model Learning

The actor and the executor form a conditional generative process that translates the current state x_t to target y. Instead of learning the generative model alone like a VAE, we maximize...
the conditional likelihood following the stochastic policy \(\pi_{\phi}\):

\[
p(y|x_t) = \int \pi_{\phi}(z_t|x_t)\eta_{\phi}(y|x_t, z_t)dz_t.
\]

(2)

The empirical objective is to minimize the reconstruction error over all samples:

\[
L_{rec} = \mathbb{E}_{x_t, y \sim D}[\|\|T(x_t), y\|_d],
\]

(3)

where \(D\) is a replay pool, \(\|\|_d\) denotes some distance measures, such as \(L_1\) and \(L_2\). To synthesis more realistic images, we also extend the actor-executor to a high quality generative model by jointly training an adversarial loss with discriminator \(D\):

\[
L_{adv} = \mathbb{E}_{x_t, y \sim D}[\log(D(y)) + \log(1 - D(T(x_t)))]
\]

(4)

Our final DL guided objective can be expressed as

\[
L_{DL} = \lambda_{rec}L_{rec} + \lambda_{adv}L_{adv},
\]

(5)

where \(\lambda_{rec}\) and \(\lambda_{adv}\) are used to balance the reconstruction and adversarial learning. In general, the DL pathway is very flexible and can readily leverage any other advanced DL losses or techniques into this framework.

**MERL in I2IT**

In RL pathway, the rewards and the soft Q values are used to iteratively guide the stochastic policy improve. Moreover, the latent-action \(z_t\), is used to estimate soft state-action value for encouraging high-level policies. Let actor and critic with parameters \(\phi\) and \(\theta\) be the function approximators for the policy \(\pi_{\phi}\) and the soft Q-function \(Q_{\theta}\), respectively. The critic parameters \(\theta\) are trained to minimize the soft Bellman residual:

\[
J_Q(\theta) = \mathbb{E}_{(x_t, z_t) \sim D}\left[\frac{1}{2}(Q_{\theta}(x_t, z_t) - \hat{Q}_{\theta}(x_t, z_t))^2\right],
\]

(6)

\[
\hat{Q}_{\theta}(x_t, z_t) = r_t + \gamma \mathbb{E}_{x_{t+1}, z_{t+1} \sim p}[V_{\theta}(x_{t+1})],
\]

where \(V_{\theta}(x_t)\) is the soft state value function which can be computed by

\[
V_{\theta}(x_t) = \mathbb{E}_{z_t \sim \pi_{\phi}}[Q_{\theta}(x_t, z_t) - \alpha \log \pi_{\phi}(z_t|x_t)].
\]

(7)

We use a target network \(\hat{Q}_{\theta}\) to stabilize training, whose parameters \(\hat{\theta}\) are obtained by an exponentially moving average of parameters of the critic network: \(\hat{\theta} \leftarrow \tau \theta + (1 - \gamma) \hat{\theta}\). The actor parameters \(\phi\) are optimized to learn the policy towards the exponential of the soft Q-function:

\[
J_{\pi}(\phi) = \mathbb{E}_{x_t, z_t \sim D}\left[\mathbb{E}_{z_t \sim \pi_{\phi}}[\alpha \log(\pi_{\phi}(z_t|x_t)) - Q_{\theta}(x_t, z_t)]\right]
\]

(8)

In practice, we use two critics with the same structure but different parameters, \((\theta_1, \theta_2)\), to mitigate positive bias in policy improvement and accelerate training. Moreover, we automatically tune the temperature hyperparameter \(\alpha\) by

\[
J(\alpha) = \mathbb{E}_{z_t \sim \pi_{\phi}}[-\alpha \log \pi_{\phi}(z_t|x_t) - \alpha \bar{H}],
\]

(9)

where \(\bar{H}\) represents a constant entropy that equals to the negative of the dimension of the latent action spaces.

The complete algorithm of SAEC is described in Algorithm 1. The training process alternates between learning I2IT from DL pathway and policy control from RL pathway. The joint training of DL and RL ensures a fast convergence since the supervised or unsupervised loss is a “highway” to facilitate back-propagation of the gradients to the actor and the executor. As the latent-action \(z_t\) is concatenated into critic (shown in Figure 1(b)), the RL objective and DL objective can cooperate to train the stochastic actor fast and stable.

**4 Experiments**

In this section, we demonstrate experimentally the effectiveness and robustness of the SAEC framework on several different tasks of I2IT.

**4.1 Face Inpainting**

**Settings** In this experiment, we apply SAEC to the problem of face inpainting, which is to fill in pixels in the central area of a face image with synthesized contents that are semantically consistent with the original face and at the same time visually realistic. Celeba-HQ dataset is used in this study, of which 28,000 images are used for training and 2,000 images are used for testing. All images have a missing part of size of 64 × 64 cropped in the center. We compare the SAEC with several recent face inpainting methods, including CE [Pathak et al., 2016], CA [Yu et al., 2018], PEN [Zeng et al., 2019], PIC [Zheng et al., 2019] and RN [Yu et al., 2020].

The actor-executor for our methods uses the same network structure as the encoder-decoder for CE. The network structures of the discriminator also comes from CE but with a different SNGAN loss. Following the previous work [Pathak et al., 2016; Yu et al., 2018; Zheng et al., 2019], we use PSNR and SSIM as the evaluation metrics.

**Results and Analysis** As shown in Table 1, by using \(L_1\) and adversarial loss in the DL guided learning, our method achieves the best PSNR and SSIM scores comparing with the existing state-of-the-art methods. And as mentioned in the Method section, the reward function is very flexible for our RL framework. Both the PSNR and SSIM based reward are suitable for SAEC and can improve the performance on face inpainting. The qualitative comparison shown in Figure 2 illustrates that the SAEC gives obvious visual improvement for synthesizing realistic faces. The results of SAEC are very reasonable, and the generated faces are sharper and more natural. This may attribute to the high-level action \(z_t\), which focuses on learning the global semantic structure and then directs the executor with DL guided learning to further improve the local details of the generated image. We can also see that the synthesized images of the SAEC could have very different appearances from the ground truth, which indicates that although our training is based on paired images, the SAEC can successfully explore and exploit data for producing diverse results.

**Ablation Study** To illustrate the stability of training GANs in SAEC, we jointly use \(L_1\) and several advanced GAN techniques i.e. WGAN-GP [Gulrajani et al., 2017], RaGAN...
Figure 2: Visual comparison of face inpainting over all methods. Our SAEC use SNGAN for auxiliary DL guided learning. # indicates what reward is used for RL training. Our results have better visual quality even for the large pose face.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE [Pathak et al., 2016]</td>
<td>26.949</td>
<td>0.870</td>
</tr>
<tr>
<td>CA [Yu et al., 2018]</td>
<td>26.608</td>
<td>0.871</td>
</tr>
<tr>
<td>PIC [Zheng et al., 2019]</td>
<td>26.307</td>
<td>0.893</td>
</tr>
<tr>
<td>PEN [Zeng et al., 2019]</td>
<td>26.391</td>
<td>0.883</td>
</tr>
<tr>
<td>RN [Yu et al., 2020]</td>
<td>26.932</td>
<td>0.892</td>
</tr>
<tr>
<td>SAEC (SNGAN + PSNR reward)</td>
<td>27.176</td>
<td>0.882</td>
</tr>
<tr>
<td>SAEC (SNGAN + SSIM reward)</td>
<td>27.327</td>
<td>0.896</td>
</tr>
</tbody>
</table>

Table 1: Quantitative results of all methods on Celeba-HQ.

Table 2: Quantitative results of the variants methods on Celeba-HQ testing dataset (all trained with PSNR reward).

Quantitative Evaluation The quantitative results are shown in Table 3. With a similar network structure, the proposed method significantly outperforms the pix2pix and PAN model on PSNR, SSIM and LPIPS over all datasets and tasks. The SAEC even achieves a comparable or better performance than the high-quality pix2pixHD and DRPAN model, which have much more complex architectures and training strategies. Moreover, using MERL instead of PPO obviously improves performance on most tasks. These experiments illustrate that the proposed SAEC is a robust and effective solution for I2IT.

Qualitative Evaluation The qualitative results of our SAEC with other I2IT methods on different tasks are shown in Figure 3. We observe that the pix2pix and PAN sometimes suffer from mode collapse and yield blurry outputs. The pix2pixHD is unstable in different datasets especially on Facades and Cityscapes. The DRPAN is more likely to produce blurred artifacts in several parts of the predicted image on Cityscapes. In contrast, the SAEC produces more stable and realistic results. Using stochastic policy and MERL helps to explore more possible solutions so as to seek out the best generation strategy by trial-and-error in the training steps, leading to a more robust agent for different datasets and tasks.

Comparison with other RL Algorithms The key components of SAEC are substituted by other structures or other state-of-the-art RL algorithms to test their importance. We use DDPG and PPO to demonstrate the effectiveness of stochastic policy and maximum entropy RL, respectively. The learning curves of different variants on four tasks are shown in Figure 4. By using the stochastic policy and the maximum entropy framework, the training is significantly improved.
Figure 3: Visual comparison of our method with pix2pix, PAN, pix2pixHD and DRPAN over all tasks.

<table>
<thead>
<tr>
<th>Method</th>
<th>Facades label→image</th>
<th>Cityscapes image→label</th>
<th>Cityscapes label→image</th>
<th>Edges→shoes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>SSIM</td>
<td>LPIPS ↓</td>
<td>PSNR</td>
</tr>
<tr>
<td>pix2pix</td>
<td>12.290</td>
<td>0.225</td>
<td>0.438</td>
<td>15.193</td>
</tr>
<tr>
<td>PAN</td>
<td>12.779</td>
<td>0.249</td>
<td>0.387</td>
<td>16.317</td>
</tr>
<tr>
<td>pix2pixHD</td>
<td>12.357</td>
<td>0.162</td>
<td>0.336</td>
<td>17.606</td>
</tr>
<tr>
<td>DRPAN</td>
<td>13.101</td>
<td>0.276</td>
<td>0.354</td>
<td>17.724</td>
</tr>
<tr>
<td>CHAN</td>
<td>13.137</td>
<td>0.231</td>
<td>0.402</td>
<td>17.459</td>
</tr>
<tr>
<td>SAECA</td>
<td>13.163</td>
<td>0.308</td>
<td>0.366</td>
<td>17.168</td>
</tr>
<tr>
<td>SAEC</td>
<td>13.178</td>
<td>0.296</td>
<td>0.324</td>
<td>17.969</td>
</tr>
</tbody>
</table>

Table 3: Quantitative results of our SAEC with other methods over all datasets. ↓ means lower is better, SAEC* means using PPO.

Figure 4: Learning curves on different I2IT tasks. SAEC performs consistently better than other modified RL algorithms.

5 Conclusion

In this paper, we present a new RL framework for high-dimensional continuous control and apply it to I2IT problems. This framework contains a stochastic actor-executor critic (SAEC) structure that handles high dimensional policy control, high-level state representation, and low-level image generation simultaneously. We train SAEC by jointly optimizing the RL and the DL objective, which provides ubiquitous and instantaneous supervision for learning I2IT. Our empirical results on several different vision applications demonstrate the effectiveness and robustness of the proposed SAEC. We will subsequently extend this work to more computer vision tasks involving with high-dimensional continuous spaces.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61602065, Sichuan province Key Technology Research and Development project under Grant 2021YFG0038.
References


