Sequential Degradation-Consistent Parameter Scheduling for Image Restoration

Syahrul Budi Rahmadan[1]★, Muhamad Rizqi Duha Pramudya[1], Ikhlasul Amal[1], Wayan Firdaus Mahmudy[1], and Novanto Yudistira[1]

Faculty of Computer Science, Universitas Brawijaya, Malang, Indonesia
{syahrul_dev@student.ub.ac.id, riskypram263@student.ub.ac.id, ikhlasulamal695@student.ub.ac.id, wayanfm@ub.ac.id, yudistira@ub.ac.id}

Abstract. Image restoration pipelines often depend on degradation-dependent parameters, such as denoising strength and data-consistency weights, that are fixed before inference or selected through computationally expensive grid search. This paper proposes a lightweight sequential parameter scheduler for degradation-consistent image restoration. The restoration process is formulated as a finite-horizon Markov decision process, where a compact Deep Q-Network selects denoising and data-consistency parameters at each restoration step based on non-reference state features extracted from the degraded observation and intermediate restoration output. The reward balances reconstruction improvement, structural similarity, runtime cost, and degradation-consistency residuals, while clean-reference metrics are used only during training and not during test inference. Experiments on Set12-style AWGN simulations under $\sigma \in \{15, 25, 50\}$ show that the proposed scheduler consistently improves degraded inputs and random sequential scheduling while substantially reducing oracle-style grid-search evaluations. However, the current policy does not yet consistently outperform stronger practical baselines such as fixed parameter scheduling and one-shot prediction. These results indicate that DQN-based sequential scheduling is a feasible and efficient mechanism for adaptive restoration, but further full-scale benchmark validation is still required.

Keywords: Image restoration · Reinforcement learning · Parameter scheduling · Degradation consistency · Deep Q-network

1 Introduction

Image restoration aims to recover a clean image $\mathbf{x}$ from a degraded observation $\mathbf{y}$ affected by noise, blur, compression artifacts, or other acquisition degradations. CNN-based restorers such as DnCNN [11] and FFDNet [12] have shown strong performance for image denoising, while classical methods such as BM3D [2] established a strong non-local collaborative filtering baseline. Real-world denoising datasets [1,6] further highlight the gap between controlled and practical degradation conditions. Nevertheless, many practical restoration pipelines still depend on degradation-dependent parameters, such as denoising strength λ, regularization weight, or data-consistency weight w. These

★ Corresponding author.

© The Author(s) 2026
M. Murtadha et al. (eds.), *Proceedings of the 2026 11th International Conference on Sustainable Information Engineering and Technology (SIET 2026)*, Atlantis Highlights in Computer Sciences 26,
https://doi.org/10.2991/978-94-6239-828-3_4

parameters are usually fixed before inference, manually tuned, or selected through validation grid search. Such static parameter selection may become suboptimal when degradation severity varies across images or when the restoration state changes across iterative refinement steps.

A natural way to improve adaptability is to select restoration parameters based on the current image condition. However, selecting only one parameter setting at the beginning of restoration remains a one-shot decision. It does not fully exploit the sequential nature of iterative restoration, where the intermediate image evolves after each update. In early steps, stronger correction may be needed to remove dominant degradation, while in later steps, more conservative updates may be preferable to preserve edges and structural details. This motivates a sequential decision-making formulation in which a policy observes the current restoration state and updates the parameter choice over a finite restoration horizon.

Reinforcement learning provides a suitable framework for such sequential decision problems. Deep Q-Networks (DQN) have been widely used for learning value-based policies in discrete action spaces [5], while policy-gradient methods such as PPO provide an alternative for more general control settings [9]. More broadly, deep generative approaches based on latent diffusion [8] and score-based stochastic differential equations [10] have demonstrated the power of learned iterative refinement for image synthesis. Transformer-based architectures [3] have also been adopted in restoration tasks, further motivating the need for adaptive parameter control in iterative pipelines. Physics-informed learning frameworks [7,4] additionally show that incorporating domain knowledge constraints, such as degradation-consistency residuals, can improve model generalization. In this work, we focus on a compact DQN scheduler because the restoration parameter space can be discretized into a small set of degradation-relevant actions. Unlike a contextual-bandit or one-shot predictor that selects an action only from the initial state, the proposed scheduler performs closed-loop parameter selection by observing the intermediate restoration state at each step.

This paper proposes a sequential degradation-consistent parameter scheduling framework for image restoration. The restoration process is formulated as a finite-horizon Markov decision process (MDP), where the scheduler selects a combination of denoising strength and data-consistency weight at each step. The scheduler is implemented as a lightweight DQN that operates on a compact non-reference state representation, including noise-related indicators, data-consistency residual, normalized step index, proxy image-quality features, and previous action information. The proposed module is designed as an add-on scheduler for a restoration operator rather than as a new restoration backbone.

The main contributions of this paper are summarized as follows:

1. We formulate degradation-consistent parameter scheduling for image restoration as a finite-horizon MDP, enabling sequential K-step parameter decisions instead of one-shot parameter selection.
2. We design a compact DQN-based scheduler with a multi-objective reward that balances reconstruction improvement, structural similarity, runtime cost, and consistency with the degradation model.

3. We provide empirical validation against degraded input, fixed parameter scheduling, random sequential scheduling, validation-grid scheduling, one-shot prediction, and oracle-grid reference, showing that the proposed scheduler consistently improves degraded and random baselines while reducing the cost of oracle-style grid search.

2 Proposed Method

2.1 Overview

The proposed framework aims to learn a sequential policy for selecting restoration parameters during iterative image restoration. Instead of using a fixed denoising strength or selecting a single parameter before inference, the scheduler observes the current restoration state and selects a parameter action at each step. The method is designed as a lightweight scheduling module that can be attached to a restoration operator, rather than as a new image restoration backbone.

Given a degraded observation $\mathbf{y}$, the restoration process starts from $\hat{\mathbf{x}}_0 = \mathbf{y}$. At each step t, the scheduler selects an action a_t that controls the restoration operator R. The intermediate restored image is updated as

$$\hat{\mathbf{x}}_t = R(\hat{\mathbf{x}}_{t-1}, \mathbf{y}, a_t), \tag{1}$$

where $\hat{\mathbf{x}}_{t-1}$ is the previous estimate and a_t denotes the selected parameter action. This formulation allows the restoration parameters to change across the restoration horizon, enabling closed-loop adaptation based on the current intermediate output.

2.2 Degradation Model and Restoration Objective

The degraded image is modeled as

$$\mathbf{y} = F(\mathbf{x}) + \boldsymbol{\epsilon}, \tag{2}$$

where $\mathbf{x}$ is the clean image, F is the known degradation operator, and $\boldsymbol{\epsilon}$ denotes additive noise. In the AWGN denoising setting used in this study, F is the identity operator and $\boldsymbol{\epsilon}$ is Gaussian noise. The objective is to obtain a restored image $\hat{\mathbf{x}}_t$ that improves perceptual and pixel-level quality while remaining consistent with the observed degradation model.

The data-consistency residual is defined as

$$d_t = \|F(\hat{\mathbf{x}}_t) - \mathbf{y}\|_2^2 . \tag{3}$$

This term is used to discourage restoration outputs that deviate excessively from the observed degraded image under the assumed degradation model.

2.3 MDP Formulation

The sequential parameter scheduling problem is formulated as a finite-horizon Markov decision process (MDP). At each step, the agent observes a compact state, selects an action from a discrete parameter set, applies the restoration operator, and receives a reward that measures the quality and cost of the update.

State. The state s_t is represented as a 25-dimensional non-reference feature vector. It includes an estimated noise indicator, the data-consistency residual, normalized step index t/K, proxy image-quality features, and a one-hot encoding of the previous action. These features are computed from the degraded image, the intermediate restored image, the degradation model, and the previous action history.

Action. The action space is defined as the Cartesian product of denoising strength λ and data-consistency weight w:

$$\lambda \in \{0.1, 0.3, 0.5, 0.7, 1.0\}, \quad w \in \{0, 0.1, 0.5, 1.0\}. \tag{4}$$

An additional stop action is included to allow early termination. Therefore, the total number of discrete actions is $5 \times 4 + 1 = 21$.

Reward. The reward is designed to balance reconstruction quality, structural preservation, runtime cost, and degradation consistency:

$$\begin{aligned} r_t = {} & \alpha \Delta\mathrm{PSNR}_t + \beta \Delta\mathrm{SSIM}_t \\ & - \gamma p_{\mathrm{run}} - \delta \left\| F(\hat{\mathbf{x}}_t) - \mathbf{y} \right\|_2^2 . \end{aligned} \tag{5}$$

Here, $\Delta\mathrm{PSNR}_t$ and $\Delta\mathrm{SSIM}_t$ denote the improvement over the previous restoration step, p_{run} penalizes the computational cost of applying an additional restoration step, and the last term penalizes degradation-inconsistent outputs. The coefficients α, β, γ, and δ are selected empirically on the validation configuration.

Clean-reference metrics such as PSNR and SSIM are used only during training to compute the reward. During test inference, the policy does not use the clean image, PSNR, or SSIM. It only uses observable state features derived from $\mathbf{y}, \hat{\mathbf{x}}_t, F$, and previous actions.

2.4 DQN Scheduler

The scheduling policy is implemented as a compact Deep Q-Network (DQN). The network receives the state vector $s_t \in \mathbb{R}^{25}$ and predicts the Q-value for each discrete action. The architecture is defined as:

$$25 \to 128 \to 128 \to 21, \tag{6}$$

where the two hidden layers use ReLU activation functions and the output layer corresponds to the 21 discrete actions. Training follows the standard DQN procedure with an experience replay buffer, mini-batch updates, target network updates, ε-greedy exploration, discount factor 0.95, and Adam optimization. At inference time, the scheduler selects the action with the highest predicted Q-value:

$$a_t = \arg\max_a Q(s_t, a). \tag{7}$$

The restoration process continues until the maximum horizon K is reached or the stop action is selected.

2.5 Sequential Scheduling Procedure

Algorithm 1. Sequential Parameter Scheduling

Input: Degraded image $\mathbf{y}$, forward degradation operator F, and maximum restoration horizon K.
Output: Final restored image $\hat{\mathbf{x}}_t$.

1. Initialize the restoration output as $\hat{\mathbf{x}}_0 \leftarrow \mathbf{y}$ and extract the initial state s_0.
2. For each restoration step $t = 1, \ldots, K$, perform Steps 2.1–2.4.
2.1 Select the restoration action using the trained DQN policy: $a_t \leftarrow \arg\max_a Q(s_t, a)$.
2.2 If a_t corresponds to the stop action, terminate the restoration process.
2.3 Apply the restoration operator using the selected action: $\hat{\mathbf{x}}_t \leftarrow R(\hat{\mathbf{x}}_{t-1}, \mathbf{y}, a_t)$.
2.4 Extract the next state s_{t+1} from $\mathbf{y}$, $\hat{\mathbf{x}}_t$, F, and a_t.
3. Return the final restored image $\hat{\mathbf{x}}_t$.

3 Experiments and Results

3.1 Dataset and Degradation

The experiments are conducted using a Set12-style AWGN denoising simulation. The current progress evaluation uses six grayscale test images from the available fallback image-loading pipeline. The overall framework of the proposed method is illustrated in Fig. 1. Therefore, the results are reported as progress-stage validation and should not yet be interpreted as final official Set12 or BSD68 benchmark results.



Fig. 1. Overview of the proposed sequential degradation-consistent parameter scheduling framework.

The degraded image is generated by adding Gaussian noise to the clean image:

$$\mathbf{y} = \mathbf{x} + \boldsymbol{\epsilon}, \tag{8}$$

where $\mathbf{x}$ is the clean image, $\mathbf{y}$ is the noisy observation, and $\boldsymbol{\epsilon}$ denotes additive white Gaussian noise. Three noise levels are evaluated, namely $\sigma \in \{15, 25, 50\}$, representing light, moderate, and severe degradation conditions.

The restoration horizon is evaluated using $K = 5$ and $K = 10$. The $K = 5$ setting examines whether the scheduler can achieve effective restoration with fewer steps, while

the $K = 10$ setting evaluates whether a longer sequential horizon provides additional benefit. All methods are tested on the same degraded observations to ensure a fair comparison across scheduling policies.

3.2 Main Results

The main quantitative results are summarized in Table 1. The comparison is focused on the $K = 5$ setting because it provides a compact restoration horizon and avoids unnecessary additional restoration calls. The results are reported under three AWGN noise levels, namely $\sigma = 15$, $\sigma = 25$, and $\sigma = 50$.

Table 1. Main quantitative results on Set12-style AWGN with $K = 5$.

σ	Method	PSNR	SSIM	DC-Error	Calls	ms/img
15	Degraded Input	24.717 ± 0.067	0.651 ± 0.131	0.000000	0	0.00
15	Fixed Parameter	29.392 ± 0.879	0.874 ± 0.038	0.002786	5	31.11
15	Random Sequential	26.738 ± 1.646	0.719 ± 0.132	0.000501	5	22.29
15	Validation Grid	29.158 ± 0.918	0.878 ± 0.033	0.003093	5	37.09
15	One-Shot Predictor	29.502 ± 0.759	0.868 ± 0.031	0.002230	5	13.25
15	**Proposed DQN**	28.847 ± 0.960	0.878 ± 0.029	0.003342	5	25.40
15	Oracle Grid Ref.	29.571 ± 0.719	0.867 ± 0.039	0.002267	100	327.13
25	Degraded Input	20.462 ± 0.105	0.512 ± 0.137	0.000000	0	0.00
25	Fixed Parameter	26.247 ± 0.921	0.811 ± 0.036	0.009054	5	9.97
25	Random Sequential	23.353 ± 2.211	0.611 ± 0.161	0.001910	5	7.26
25	Validation Grid	26.325 ± 0.907	0.811 ± 0.037	0.008796	5	9.25
25	One-Shot Predictor	26.806 ± 0.703	0.780 ± 0.070	0.006673	5	11.90
25	**Proposed DQN**	26.423 ± 0.850	0.798 ± 0.049	0.007313	5	23.36
25	Oracle Grid Ref.	26.768 ± 0.731	0.800 ± 0.050	0.007211	100	217.55
50	Degraded Input	15.011 ± 0.138	0.327 ± 0.104	0.000000	0	0.00
50	Fixed Parameter	23.190 ± 0.648	0.688 ± 0.057	0.028987	5	8.70
50	Random Sequential	19.095 ± 2.785	0.464 ± 0.170	0.008766	5	8.28
50	Validation Grid	22.654 ± 0.810	0.695 ± 0.027	0.030868	5	9.18
50	One-Shot Predictor	23.164 ± 0.684	0.683 ± 0.054	0.027615	5	8.34
50	**Proposed DQN**	22.200 ± 0.866	0.651 ± 0.040	0.027527	5	14.26
50	Oracle Grid Ref.	23.058 ± 0.638	0.687 ± 0.051	0.027864	100	177.64

The proposed DQN scheduler consistently improves the degraded input across all noise levels. At $\sigma = 15$, the PSNR increases from 24.717 dB to 28.847 dB. At $\sigma = 25$, it increases from 20.462 dB to 26.423 dB. Under the most severe noise level, $\sigma = 50$, the DQN scheduler improves PSNR from 15.011 dB to 22.200 dB. These results show that the learned scheduler can provide meaningful restoration improvement under light, moderate, and severe AWGN degradation.

The proposed scheduler also consistently outperforms Random Sequential. The PSNR gains over Random Sequential are 2.109 dB, 3.070 dB, and 3.105 dB for $\sigma = 15$, $\sigma = 25$, and $\sigma = 50$, respectively. This comparison is important because both methods apply multi-step restoration. Therefore, the improvement does not simply come from applying several restoration steps, but from using a learned action-selection policy.

However, the DQN scheduler does not consistently outperform stronger practical baselines. At $\sigma = 25$, the proposed scheduler is competitive and obtains higher PSNR

than Fixed Parameter and Validation Grid. Nevertheless, the One-Shot Predictor and Oracle Grid Reference still obtain higher PSNR. At $\sigma = 15$ and $\sigma = 50$, the DQN scheduler improves over Degraded Input and Random Sequential, but remains below Fixed Parameter and One-Shot Predictor in PSNR. Therefore, the current result should be interpreted as evidence that DQN-based sequential scheduling is feasible and useful, but not yet as evidence that it dominates all practical baselines.

In terms of computational cost, the DQN scheduler uses only 5 restorer calls, whereas the Oracle Grid Reference uses 100 calls. This corresponds to a 95% reduction in restorer evaluations. The DQN scheduler is also substantially faster than the oracle reference. However, it is not always faster than simpler practical baselines because state extraction and policy inference introduce additional overhead.

3.3 Effect of Restoration Horizon

To examine whether a longer restoration horizon improves the learned policy, the DQN scheduler is evaluated with $K = 5$ and $K = 10$. The comparison is shown in Table 2. This analysis focuses on the proposed DQN scheduler because the goal is to understand whether additional sequential decisions improve the proposed policy.

Table 2. Effect of restoration horizon on the proposed DQN scheduler.

σ	PSNR K=5	PSNR K=10	ΔPSNR	SSIM K=5	SSIM K=10	Calls
15	28.847	28.248	−0.599	0.878	0.877	5/10
25	26.423	26.416	−0.007	0.798	0.790	5/10
50	22.200	21.587	−0.613	0.651	0.669	5/10

The comparison shows that increasing the horizon from $K = 5$ to $K = 10$ does not produce a clear PSNR improvement. At $\sigma = 15$, PSNR decreases by 0.599 dB. At $\sigma = 25$, the difference is very small, but the $K = 10$ result is still slightly lower. At $\sigma = 50$, PSNR decreases by 0.613 dB, although SSIM slightly increases. These results suggest that a longer horizon is not automatically beneficial. Additional restoration steps can increase the number of restorer calls and may introduce over-processing when the policy is not yet fully stable. Therefore, $K = 5$ is used as the more efficient and stable setting in the main comparison.

3.4 Ablation Study

An ablation study is conducted to analyze the effect of several design choices, including action granularity, restoration horizon, reward formulation, state representation, and stop action. Table 3 summarizes the available ablation results under the severe AWGN configuration ($\sigma = 50$).

The ablation results show that the restoration horizon has a strong influence on reconstruction quality. The $K = 1$ setting performs poorly because it does not provide sufficient iterative refinement. Increasing the horizon to $K = 3$ improves PSNR substantially, while $K = 5$ provides the best result in this configuration. However, extending

Table 3. Ablation study under the severe AWGN configuration ($\sigma = 50$).

Group	Variant	PSNR	SSIM	DC-Error
Action space	Small action space	21.393 ± 1.032	0.650 ± 0.010	0.035030
Action space	Current 21 actions	21.139 ± 1.102	0.646 ± 0.010	0.034558
Horizon	$K = 1$	14.995 ± 0.114	0.324 ± 0.106	0.000000
Horizon	$K = 3$	21.251 ± 0.239	0.526 ± 0.127	0.013286
Horizon	$K = 5$	22.304 ± 0.871	0.681 ± 0.018	0.032593
Horizon	$K = 10$	21.786 ± 0.984	0.673 ± 0.007	0.032705
Reward	PSNR-only	21.177 ± 1.352	0.611 ± 0.015	0.032140
Reward	PSNR+SSIM	21.747 ± 0.960	0.633 ± 0.035	0.027720
Reward	No DC penalty	21.361 ± 1.045	0.655 ± 0.008	0.034222
Reward	Full reward	21.457 ± 0.804	0.659 ± 0.006	0.030953
State	No previous action	22.025 ± 0.917	0.639 ± 0.044	0.026494
State	No DC feature	21.420 ± 1.100	0.652 ± 0.014	0.031602
State	Full state	21.139 ± 1.102	0.646 ± 0.010	0.034558
Stop action	Without stop	22.266 ± 0.852	0.643 ± 0.049	0.048580
Stop action	With stop	21.457 ± 0.804	0.659 ± 0.006	0.030953

the horizon to $K = 10$ reduces PSNR, indicating diminishing returns and possible over-processing.

The reward ablation shows that adding structural and consistency-related terms changes the quality trade-off. The full reward does not obtain the highest PSNR, but it provides a more balanced SSIM and data-consistency behavior than the PSNR-only reward. This suggests that the proposed reward is not purely optimized for PSNR, but attempts to balance reconstruction quality and degradation consistency.

The state and stop-action ablations show mixed behavior. Removing previous action information or disabling the stop action can improve PSNR in this progress run, but these variants do not necessarily improve the overall balance between SSIM and data-consistency error. Therefore, the ablation results should not be interpreted as final evidence that the current full configuration is optimal. Instead, they indicate that the state representation and stop-action design still require further refinement in the full benchmark experiment.

3.5 Qualitative Analysis

A qualitative comparison is provided in Fig. 2 to inspect the visual behavior of the restoration outputs. The comparison uses a Set12-style test image under severe AWGN degradation ($\sigma = 50$).

The degraded image contains strong granular noise that obscures both smooth regions and structural boundaries. The fixed and validation-grid baselines remove a substantial amount of noise, but may also smooth fine details. The DQN scheduler also reduces the visible noise and preserves several major structures, although it does not yet provide the strongest quantitative result under the severe-noise setting. This observation is consistent with the quantitative results in Table 1: the proposed scheduler clearly improves the degraded input and random scheduling, but it does not yet consistently outperform the strongest practical baselines.

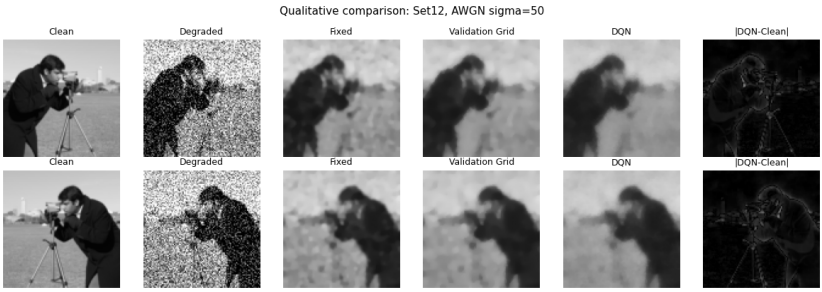


Fig. 2. Visual comparison of restoration outputs under severe AWGN degradation ($\sigma = 50$). From left to right: degraded input, fixed parameter baseline, and proposed DQN scheduler output.

3.6 Discussion

The results indicate three main findings. First, the proposed DQN scheduler is effective in improving degraded inputs across all tested noise levels. This shows that the learned policy can select useful restoration actions rather than applying uncontrolled random steps.

Second, the DQN scheduler is computationally more efficient than oracle-style grid search. The oracle reference requires 100 or 200 restorer evaluations, whereas the proposed scheduler requires only 5 or 10 calls. This supports the main motivation of sequential scheduling as a lightweight alternative to expensive grid-search-based parameter selection.

Third, the current DQN policy is not yet consistently better than stronger practical baselines such as Fixed Parameter and One-Shot Predictor. The most competitive result appears at $\sigma = 25$, where the proposed scheduler outperforms Fixed Parameter and Validation Grid in PSNR. However, under light and severe noise, simpler baselines remain stronger. This suggests that further training, action-space refinement, and reward tuning are needed before the method can be claimed to outperform practical non-oracle baselines more consistently.

4 Conclusion

This paper presented a sequential degradation-consistent parameter scheduling framework for image restoration. The method formulates parameter selection as a finite-horizon MDP and uses a compact DQN scheduler to select denoising and data-consistency parameters at each restoration step. The scheduler is designed as a lightweight add-on module and does not require clean-reference information during test inference.

The progress-stage experiments on Set12-style AWGN simulations show that the proposed DQN scheduler consistently improves degraded inputs and random sequential scheduling under $\sigma \in \{15, 25, 50\}$. The method also reduces the number of oracle-style grid-search evaluations by approximately 95% in the $K = 5$ setting. However, the current policy does not yet consistently outperform stronger practical baselines such as Fixed Parameter and One-Shot Predictor. The horizon analysis further shows that

increasing the horizon from $K = 5$ to $K = 10$ does not necessarily improve PSNR and may introduce diminishing returns.

For future work, the evaluation should be completed using the official Set12 and BSD68 benchmarks, and a stronger denoising baseline such as BM3D or FFDNet should be included. Further improvements should also focus on stabilizing the learned policy, refining the action space, and improving the reward design so that sequential scheduling can more consistently outperform practical non-oracle baselines.

Acknowledgments. This research was supported by the Faculty of Computer Science, Universitas Brawijaya, Malang, Indonesia.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2912–2920. IEEE (2018)
2. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Transactions on Image Processing **16**(8), 2080–2095 (2007)
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR) (2021)
4. Li, L., Wang, W., Ong, Y.S.: Physics-guided machine learning for scientific discoveries. Nature Machine Intelligence **4**(9), 784–796 (2022)
5. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
6. Plötz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1586–1595. IEEE (2017)
7. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear PDEs. Journal of Computational Physics **378**, 686–707 (2019)
8. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10684–10695. IEEE (2022)
9. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
10. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: International Conference on Learning Representations (ICLR) (2021)
11. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. IEEE Transactions on Image Processing **26**(7), 3142–3155 (2017)
12. Zhang, K., Zuo, W., Zhang, L.: FFDNet: Toward a fast and flexible solution for CNN-based image denoising. IEEE Transactions on Image Processing **27**(9), 4608–4622 (2018)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

