Universal Compressed Image Restoration via Codec-Aware Conditioning with Reinforcement Learning

Changwoo Han[1*], Hongil Kim[1*], Donghyun Kim[2], Sung-Chang Lim[2], Seung-Won Jung[1†]

[1]Department of Electrical Engineering, Korea University, Seoul, Korea
[2]Media Coding Research Section, Electronics and Telecommunications Research Institute, Daejeon, Korea
{hcwoo329, yourhong1, swjung83}@korea.ac.kr, {kimddng, sclim}@etri.re.kr

Abstract

We address the task of universal compressed image restoration, which involves recovering high-quality images degraded by a wide range of codecs and compression levels. While prior methods have made significant progress, they typically target specific degradation types and struggle to generalize across both traditional and learning-based codecs. To overcome this limitation, we propose a unified framework that leverages codec-aware conditioning and reinforcement learning-based fine-tuning. Specifically, we introduce a conditioning module that encodes both codec type and compression level, enabling the restoration network to adapt its behavior to diverse degradation settings. To further improve generalization, we incorporate reward-based objectives during fine-tuning, providing complementary signals that enhance training across both conventional and learned compression schemes. Experimental results demonstrate the effectiveness of our method in restoring images across a wide range of compression artifacts and scenarios.

Introduction

Lossy compression techniques are widely used across most visual media applications. These include traditional codec standards, such as JPEG (Wallace 1991), HEVC (Sullivan et al. 2012), VVC (Bross et al. 2021), and WebP (Ginesu, Pintus, and Giusto 2012), which rely on hand-designed transforms and block-based quantization, and end-to-end learned codecs (Cheng et al. 2020; Mentzer et al. 2020) that use neural networks trained for rate-distortion optimization through learned transforms and entropy models. While these methods enable efficient storage and transmission, aggressive compression, particularly at low bitrates, often exacerbates visual quality, introducing artifacts such as blocking, blurring, color shifts, and loss of fine textures. The field of compressed image restoration (CIR) aims to mitigate these artifacts and reconstruct high-fidelity images. However, most CIR models are trained for specific codecs or fixed compression levels, which limits their ability to generalize in real-world scenarios where both the codec type and compression severity can vary.

Training a single restoration model to handle multiple types of compression artifacts often leads to gradient conflict (Yu et al. 2020; Zhu et al. 2022; Yang et al. 2024; Tian et al. 2025). For example, optimizing for blocking artifact removal may conflict with adjustments needed to reduce blurring. To mitigate this issue, prior works have proposed adapting the model parameters or feature representations according to the degradation type. These approaches emphasize the importance of providing explicit degradation information to guide the restoration process effectively.

Recent methods follow this idea but differ in how they obtain and incorporate degradation-specific knowledge during restoration. Some methods rely on internally learned prompts or soft condition tokens to modulate the restoration network (Potlapalli et al. 2023; Li et al. 2024a). However, their implicit degradation representations make it challenging for the model to adapt to previously unseen degradations. Other methods introduce auxiliary modules to predict compression parameters such as quality factors (QFs) (Kim, Soh, and Cho 2020; Jiang, Zhang, and Timofte 2021) or quantization parameters (QPs) (Kim et al. 2020; Ding et al. 2024). But these explicit degradation representations are codec-specific, limiting their applicability across different codecs. Consequently, extending them to support a new codec typically requires significant structural modifications and re-training, which limits their scalability and forward-compatibility.

In this work, we present a reinforcement learning (RL)-based universal compressed image restoration (UCIR) framework, which decouples degradation modeling from image reconstruction. Specifically, considering that codec types and compression levels affect the type and severity of compression artifacts, we design a policy network (P-Net) that predicts these conditions directly from compressed images. By incorporating the predicted conditions as inputs, our proposed conditional CIR network (R-Net) restores compressed images across various codec types and degradation levels. Unlike previous prompt-based approaches with implicit representations or designs tied to a single codec, our framework enables adaptive restoration by conditioning the network on the predicted codec type and compression level. This design allows a single restoration model to adapt across diverse compression settings without structural changes.

To further enhance generalization, we adopt a two-stage

[*]These authors contributed equally.
[†]Corresponding author.

training strategy. In the first stage, P-Net and R-Net are trained separately using their respective supervised losses: P-Net is trained to predict codec type and compression level from compressed images, while R-Net is trained to restore images conditioned on ground-truth degradation information. In the second stage, the two networks are jointly fine-tuned. Since the number of possible codec types and compression levels can be prohibitively large, we replace ground-truth labels with reward-based learning, allowing the model to generalize to diverse compression conditions without requiring explicit annotations. P-Net is optimized using a hybrid loss that combines reconstruction loss and reinforcement learning objectives, where the RL component is computed from perceptual metrics such as the structural similarity index measure (SSIM) reward. This reinforcement signal enables P-Net to explore and refine its condition predictions beyond the limitations of supervised learning. Our learning strategy is aligned with recent findings emphasizing the importance of task-aware optimization in CIR (Ren et al. 2024). Overall, our approach achieves robust and controllable restoration by adaptively calibrating the restoration process.

The main contributions of this paper are as follows:

- We propose a UCIR framework that predicts codec type and compression level, and uses this information to guide the restoration process, offering improved interpretability and compatibility across diverse compression scenarios.
- We introduce a training strategy in which P-Net is optimized using a combination of supervised losses and perceptual rewards. This hybrid learning strategy allows R-Net to generalize across differentiable and non-differentiable codecs, including previously unseen compression configurations.
- Extensive experiments on a UCIR task have shown that our framework outperforms existing methods across a wide spectrum of compression artifacts. This strong performance validates the framework's capacity for generalization, adaptation, and scalability.

Related Work

Reinforcement Learning for Image Processing

The application of RL to image processing has advanced significantly over the past several years. Early pioneering works demonstrated its potential for a range of tasks. For instance, methods like (Yu et al. 2018) established a foundation by using an agent to select appropriate operations for restoring corrupted images. Similarly, early studies in color enhancement (Park et al. 2018; Hu et al. 2018; Cao et al. 2017) utilized agents to iteratively apply global and local adjustments. These foundational studies, including pixel-level control (Furuta, Inoue, and Yamasaki 2020), showed that RL could effectively handle complex image processing workflows.

Building on these initial successes, recent research has shifted towards developing more efficient, adaptable, and context-aware agents. For instance, in low-light image enhancement, recent methods (Zhang et al. 2021; Ogino, Toizumi, and Ito 2025) leverage RL to learn tailored enhancement policies using no-reference reward functions, offering greater flexibility than conventional supervised methods. CURVE (Ogino, Toizumi, and Ito 2025), in particular, integrates large-scale pre-trained models like CLIP (Radford et al. 2021) to provide rewards, enabling enhancement that aligns better with human perception.

Compressed Image Restoration

CIR aims to recover high-quality images from compressed inputs affected by artifacts such as blocking, blurring, and color distortion (Dong et al. 2015; Cho et al. 2019; Wang et al. 2020; Deng et al. 2020; Zhang et al. 2023). To handle varying compression degradations, some methods introduce explicit degradation predictors using auxiliary networks that estimate compression parameters like QFs (Kim, Soh, and Cho 2020; Jiang, Zhang, and Timofte 2021) or QPs (Kim et al. 2020; Ding et al. 2024). These predictors provide restoration networks with external condition signals derived from compressed inputs. Other approaches employ implicit conditioning mechanisms, where learnable prompts or soft condition tokens are used to modulate the restoration process (Potlapalli et al. 2023; Li et al. 2024a,b).

Recently, diffusion-based CIR methods (Ren et al. 2024; Guo et al. 2025) have incorporated generative priors, applying prompt or condition modulation in combination with encoder or decoder fine-tuning to enhance restoration across a wider range of degradations. MoE-DiffIR (Ren et al. 2024) further extends this direction by introducing a UCIR framework that dynamically routes task-specific prompts and adapts to diverse degradation types.

Proposed Method

In this section, we first describe R-Net, which restores images by leveraging condition information, namely the codec type and compression level. Then, we introduce P-Net, designed to predict these conditions, enabling R-Net to perform conditional image restoration. Finally, we detail the training strategy where our P-Net and R-Net are jointly trained to generalize better across diverse compression conditions.

Restoration Network

Compression artifacts vary significantly across codec types and compression levels, resulting in diverse degradation patterns. To enable R-Net to adapt to these variations, R-Net is conditioned on explicit degradation information. Specifically, R-Net employs a U-shaped encoder-decoder architecture with Transformer blocks, where its decoder part is guided by condition features generated by the codec-aware conditioning module (CCM).

As shown in Fig. 1(c), CCM encodes the codec type and compression level into a fused feature representation. Specifically, the codec type is mapped to a learnable embedding $F_t \in \mathbb{R}^{H \times W \times C_t}$, and the compression level, normalized to the range $[0, 1]$, is passed through a convolution block, resulting in $F_l \in \mathbb{R}^{H \times W \times C_l}$, where H and W denote spatial dimensions, while C_t and C_l are channel dimensions.

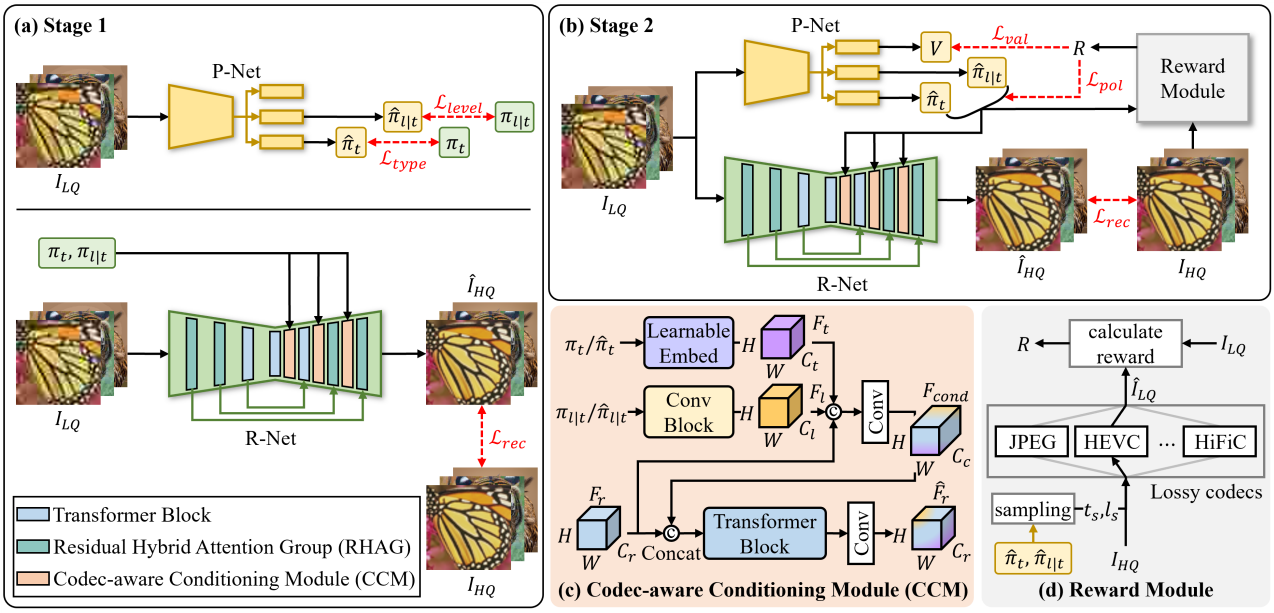

Figure 1: Overview of our proposed UCIR framework. (a) depicts Stage 1, where P-Net is trained to predict codec-aware conditions, and R-Net to restore images using ground-truth conditions. R-Net consists of Transformer blocks, the RHAGs adapted from HAT (Chen et al. 2023), and the CCMs. (b) shows Stage 2, in which P-Net and R-Net are jointly optimized. P-Net generates conditions for R-Net and is fine-tuned using reward and reconstruction losses. (c) shows the structure of CCM, which fuses the conditions and image feature, generating a fused feature to guide restoration. (d) illustrates the reward module, which computes the reward for P-Net training based on the sampled conditions.

These condition features are concatenated with the image feature, denoted as $F_r \in \mathbb{R}^{H\times W\times C_r}$, and fused by a convolution layer:

$$F_{cond} = \text{Conv}\left(\text{Concat}\left(F_r, F_t, F_l\right)\right). \tag{1}$$

The fused condition feature $F_{cond} \in \mathbb{R}^{H\times W\times C_c}$ is then concatenated with F_r, and the combined features are passed through a Transformer block, resulting in the integrated feature $\hat{F}_r$.

Policy Network

To mitigate the optimization challenges inherent in multi-degradation restoration, we introduce P-Net, a dedicated module designed to infer the codec type and compression level from a compressed input image. By decoupling degradation prediction from image restoration, P-Net improves training stability and provides an interpretable conditioning signal for R-Net.

The architecture of P-Net is tailored to extract degradation-specific information. Given an input image I_{LQ}, the network first applies a feature extractor consisting of three downsampling blocks to produce a feature map F_p. This feature map is then passed to three separate branches, which are responsible for predicting the codec type, the compression level, and a state-value, respectively.

The codec type branch, denoted as P_t, takes the shared feature map F_p as input. It processes F_p through its own network path, culminating in an MLP head with a Softmax function to produce a single probability distribution over all supported codec types:

$$\hat{\pi}_t = P_t(F_p). \tag{2}$$

The compression level branch, P_l, also processes F_p through its convolutional layers to generate compression level-specific feature vectors. Since compression levels are defined differently across codecs, P_l uses separate prediction heads, each dedicated to a specific codec. Consequently, P_l outputs a set of probability distributions, where each distribution $\hat{\pi}_{l|t}$ represents the predicted levels for its respective codec:

$$\{\hat{\pi}_{l|t}\}_{t\in T} = P_l(F_p), \tag{3}$$

where T denotes the set of all supported codec types.

Finally, the value branch, P_v, estimates the state-value V for the RL fine-tuning stage. It processes the shared feature map F_p through its network path to regress a single scalar value:

$$V = P_v(F_p). \tag{4}$$

Training Strategy

Supervised training is often insufficient to achieve robust generalization, especially when encountering novel compression characteristics or when dealing with non-differentiable codecs. To address this limitation, we adopt a two-stage training strategy. In the first stage, R-Net and P-Net are separately optimized using their respective supervised losses. In the second stage, the two networks are jointly fine-tuned while the RL loss is introduced to guide the training of P-Net.

Algorithm 1: Second-Stage Training Procedure

Input: Training dataset $\mathcal{D} = \{(I_{LQ}, I_{HQ})\}$, Total fine-tuning epochs N, Stage 1-trained P-Net P_ϕ and R-Net R_θ
Output: P-Net P_ϕ, R-Net R_θ

for epoch $e = 1$ to N **do**
  for each batch (I_{LQ}, I_{HQ}) from $\mathcal{D}$ **do**
    // Step 1: Predict and sample
    $\{\hat{\pi}_t, \hat{\pi}_{l|t}, V\} \leftarrow P_\phi(I_{LQ})$
    $\hat{I}_{HQ} \leftarrow R_\theta(I_{LQ}, \hat{\pi}_t, \hat{\pi}_{l|t})$
    $t_s \sim \hat{\pi}_t$; $l_s \sim \hat{\pi}_{l|t_s}$

    // Step 2: Calculate losses
    Compress I_{HQ} using the sampled codec type t_s with compression level l_s to obtain $\hat{I}_{LQ}$
    $R \leftarrow \text{SSIM}(\hat{I}_{LQ}, I_{LQ})$
    $A \leftarrow R - V$
    Calculate $\mathcal{L}_{total}$ using Eq. (11)

    // Step 3: Update network parameters
    Update ϕ and θ by back-propagating gradients from $\mathcal{L}_{total}$
  end for
end for

Specifically, in the first stage, R-Net is trained while taking the ground-truth codec type and compression level as conditions. These conditions are represented as one-hot vectors, denoted as π_t and $\pi_{l|t}$, which correspond to $\hat{\pi}_t$ in (2) and $\hat{\pi}_{l|t}$ in (3), respectively. The output of R-Net, $\hat{I}_{HQ}$, is compared with its corresponding ground-truth image I_{HQ} using the reconstruction loss $\mathcal{L}_{rec}$:

$$\mathcal{L}_{rec} = \left\| \hat{I}_{HQ} - I_{HQ} \right\|_1, \tag{5}$$

where $\|\cdot\|_1$ denotes the L1 norm.

In the first stage, P-Net is trained using the ground-truth conditions, π_t and $\pi_{l|t}$. Namely, the codec type branch P_t and the compression level branch P_l of P-Net are supervised using π_t and $\pi_{l|t}$, resulting in the codec type prediction loss $\mathcal{L}_{type}$ and compression level prediction loss $\mathcal{L}_{level}$, respectively. The codec type loss $\mathcal{L}_{type}$ is defined as:

$$\mathcal{L}_{type} = -\sum_i \pi_t(i) \log(\hat{\pi}_t(i)), \tag{6}$$

where $\hat{\pi}_t(i)$ represents a probability of the i-th codec type in T. The compression level loss $\mathcal{L}_{level}$ is defined as:

$$\mathcal{L}_{level} = -\sum_i \pi_{l|t^*}(i) \log(\hat{\pi}_{l|t^*}(i)), \tag{7}$$

where t^* is the ground-truth codec type, and $\hat{\pi}_{l|t^*}(i)$ is a probability of the i-th compression level of the codec type t^*. In other words, among multiple heads in the compression level branch, only the head corresponding to the codec type t^* is used for the loss measurement. The supervised loss for the conditions, $\mathcal{L}_{cond}$, is then formulated as the weighted sum of these two losses:

$$\mathcal{L}_{cond} = \lambda_{type}\mathcal{L}_{type} + \lambda_{level}\mathcal{L}_{level}, \tag{8}$$

where λ_{type} and λ_{level} are weighting coefficients. The value branch of P-Net is not trained during the first stage.

In the second stage, we jointly fine-tune P-Net and R-Net using a hybrid objective that combines a standard reconstruction loss with a reinforcement learning framework. This stage is designed to enhance the model's adaptability, particularly for diverse degradations and non-differentiable codecs. This setup avoids the need for explicit ground-truth labels in the second stage, where the model instead learns to generalize to novel degradations by leveraging reward-based feedback aligned with the type and level distributions learned in the first stage. We formulate the training process within an Actor-Critic paradigm (Mnih et al. 2016), where P-Net serves as the agent. The entire procedure for the second-stage training is summarized in Algorithm 1.

Specifically, in the second stage, R-Net is trained using the conditions predicted by P-Net, allowing it to handle diverse compression artifacts. The same reconstruction loss $\mathcal{L}_{rec}$ in (5) is used for the supervision of R-Net. P-Net then samples an action $a_s = (t_s, l_s)$, where the codec type t_s and compression level l_s are drawn from the predicted probability distributions, $\hat{\pi}_t$ and $\hat{\pi}_{l|t}$, respectively. This action is used to compress I_{HQ}, as illustrated in Fig. 1(d). For example, if JPEG with QF of 40 is selected, I_{HQ} is compressed using a standard JPEG encoder (Wallace 1991) with QF of 40, maintaining the rest of the default encoder settings. The performance of P-Net is evaluated by comparing the resulting image $\hat{I}_{LQ}$ with the input compressed image I_{LQ}, using SSIM as the reward signal, denoted as R. This guides P-Net to infer parameters that closely replicate the input compression, without relying on ground-truth degradation labels.

R-Net and P-Net are jointly optimized with a total loss $\mathcal{L}_{total}$, which is a weighted sum of three components: a reconstruction loss $\mathcal{L}_{rec}$, a policy loss $\mathcal{L}_{pol}$, and a value loss $\mathcal{L}_{val}$. The policy loss, which updates the actor component of P-Net, is defined using the advantage A, where $A = R - V$. This loss encourages actions that yield higher-than-expected rewards:

$$\mathcal{L}_{pol} = -\log p(a_s) \cdot A, \tag{9}$$

where $p(a_s)$ is the probability of the sampled action. The value loss updates the critic component of P-Net, training it to produce more accurate estimates of the expected reward:

$$\mathcal{L}_{val} = (R - V)^2. \tag{10}$$

The total loss $\mathcal{L}_{total}$ is given as:

$$\mathcal{L}_{total} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{pol}\mathcal{L}_{pol} + \lambda_{val}\mathcal{L}_{val}, \tag{11}$$

where λ_{rec}, λ_{pol}, and λ_{val} are hyper-parameters that balance the contribution of each term. The parameters of both P-Net and R-Net are updated jointly by backpropagating the gradients from this total loss.

Experiments

Datasets

We define three groups of compression settings, which we denote as D_1, D_2, and D_u. D_1 and D_2 are used for supervised training and RL-based fine-tuning, respectively, while D_u is reserved solely for evaluating generalization.

Codecs	Methods	LIVE1		Classic5		BSDS500		DIV2K		Urban100	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
JPEG	AirNet	30.51	0.875	31.98	0.872	30.42	0.872	32.38	0.896	29.53	0.904
	PromptIR	30.27	0.871	31.20	0.861	30.34	0.871	31.18	0.893	28.78	0.893
	PromptCIR	30.40	0.872	31.44	0.862	30.45	0.873	32.32	0.895	29.10	0.896
	MoE-DiffIR	27.50	0.825	29.64	0.857	27.90	0.824	27.04	0.842	26.47	0.847
	DFPIR	30.72	**0.879**	32.06	0.874	30.57	0.875	32.71	**0.901**	29.86	0.910
	Ours (Stage 1)	**30.77**	**0.879**	**32.19**	**0.875**	**30.75**	**0.879**	**32.75**	**0.901**	**30.07**	**0.912**
	Ours (Stage 2)	**30.77**	**0.879**	32.15	0.874	**30.75**	**0.879**	**32.75**	**0.901**	30.05	**0.912**
HiFiC	AirNet	28.29	0.832	29.94	0.833	28.37	0.828	30.71	0.872	26.82	0.854
	PromptIR	28.21	0.830	29.88	0.829	28.43	0.830	30.62	0.870	26.61	0.857
	PromptCIR	28.26	0.830	29.91	0.830	28.47	0.830	30.66	0.870	26.70	0.858
	MoE-DiffIR	26.56	0.789	29.15	0.842	27.05	0.792	25.85	0.800	25.28	0.814
	DFPIR	28.41	0.832	30.03	0.834	28.45	0.828	30.84	0.873	27.01	0.865
	Ours (Stage 1)	**28.45**	**0.835**	30.02	**0.835**	**28.63**	**0.835**	30.84	**0.875**	**27.09**	**0.867**
	Ours (Stage 2)	**28.45**	**0.835**	**30.05**	**0.835**	28.62	**0.835**	**30.87**	0.874	**27.09**	**0.867**

Table 1: Quantitative comparison for CIR, averaged over JPEG with QFs of [10, 20, 30, 40] and HiFiC (Mentzer et al. 2020) with ["low", "medium", "high"] configurations in D_1. Results are reported on five datasets in terms of PSNR↑/SSIM↑. The best result is in **bold**. All models are trained from scratch using the same dataset for fair comparisons. Additional results on other codecs and compression settings are provided in the supplementary material.

Methods	JPEG (QF=5, 70)		HEVC (QP=47, 22)		VVC (QP=47, 22)	
	LIVE1	DIV2K	LIVE1	DIV2K	LIVE1	DIV2K
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
AirNet	29.99 / 0.825	31.49 / 0.851	34.19 / 0.858	35.41 / 0.886	33.80 / 0.871	34.33 / 0.881
PromptIR	29.94 / 0.824	31.52 / 0.852	34.22 / 0.859	35.47 / 0.886	33.68 / 0.870	34.26 / 0.880
PromptCIR	30.05 / 0.825	31.65 / 0.854	34.21 / 0.859	35.50 / 0.887	33.78 / 0.870	34.36 / 0.881
MoE-DiffIR	26.40 / 0.770	25.90 / 0.787	27.19 / 0.779	26.68 / 0.798	27.82 / 0.813	27.31 / 0.833
DFPIR	29.00 / 0.793	30.41 / 0.816	32.81 / 0.830	34.09 / 0.862	33.04 / 0.868	33.97 / 0.879
Ours (Stage 1)	29.29 / 0.818	30.64 / 0.837	33.16 / 0.850	34.58 / 0.881	32.58 / 0.864	34.46 / 0.883
Ours (Stage 2)	**30.38 / 0.834**	**32.01 / 0.862**	**34.58 / 0.864**	**35.91 / 0.891**	**34.05 / 0.873**	**34.72 / 0.884**

Table 2: Quantitative comparison on compression levels in D_2, averaged on LIVE1 and DIV2K in terms of PSNR↑/SSIM↑. The best result is in **bold**.

The first group, D_1, is used during the first stage and contains 23 compression settings: JPEG with QFs of [10, 20, 30, 40], HEVC and VVC with QPs of [42, 37, 32, 27], C_{PSNR} and C_{SSIM} (Cheng et al. 2020) with quality levels of [1, 2, 3, 4], and HiFiC (Mentzer et al. 2020) with ["low", "medium", "high"] configurations.

The second group, D_2, is introduced in the second stage and includes 11 novel compression settings not seen in D_1, used to improve the adaptability to unlabeled conditions. These include JPEG with QFs of [5, 70], HEVC and VVC with QPs of [47, 22], and WebP with QFs of [5, 10, 20, 30, 40].

To evaluate the generalization ability, the third group, D_u, is used only at test time and contains entirely unseen settings, including AVC (Wiegand et al. 2003) with QPs of [42, 37, 32, 27], and various double-compressed combinations, including JPEG→VVC, VVC→JPEG, VVC→C_{PSNR}, and C_{PSNR}→VVC.

For the training dataset, we use 3,450 images from DF2K (Timofte et al. 2017). These images are compressed using the compression settings in D_1 and D_2. For the test dataset, we use five commonly used datasets: LIVE1 (Sheikh 2005), Classic5 (Zeyde, Elad, and Protter 2012), BSDS500 (Arbeláez et al. 2011), DIV2K validset (Agustsson and Timofte 2017), and Urban100 (Huang, Singh, and Ahuja 2015). Evaluation is conducted under three groups of settings corresponding to D_1, D_2, and D_u.

Implementation Details

For the first stage, we train R-Net for 300 epochs, following the same configuration as PromptCIR (Li et al. 2024a), with the number of blocks set to $[4, 6, 6, 8]$ and attention heads to $[1, 2, 4, 8]$. P-Net is pretrained separately for 500 epochs. Details of the training settings for each network are provided in the supplementary material.

During the second stage, we jointly optimize the networks using the loss function $\mathcal{L}_{total}$. We use the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and set the initial learning rate to 2×10^{-5}. A CosineAnnealing scheduler is used over 50 epochs. We use random cropping with a patch size of 224×224 and a batch size of 4. All experiments are conducted using 4 NVIDIA Quadro RTX 8000 GPUs.

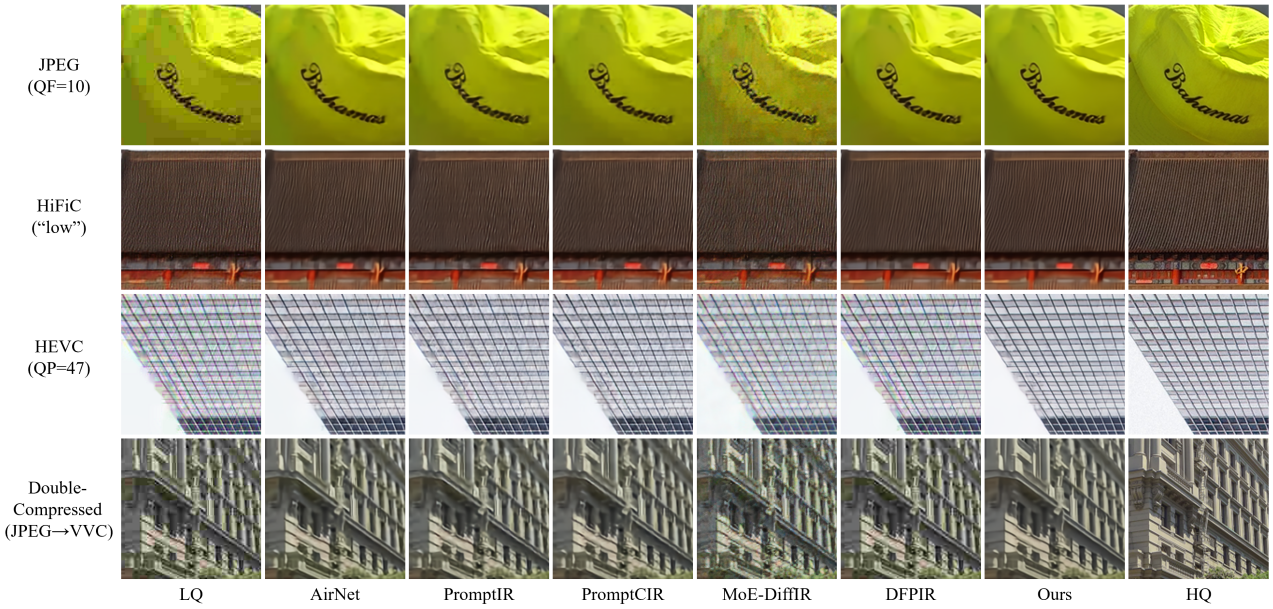

Figure 2: Visual comparison between our methods and other existing methods. Each row corresponds to a different compression task: JPEG (QF=10), HiFiC ("low"), HEVC (QP=47), and double-compressed (JPEG→VVC). More visual results can be found in the supplementary.

Methods	WebP (QF=5, 10, 20, 30, 40)	
	LIVE1	DIV2K
	PSNR/SSIM	PSNR/SSIM
AirNet	29.81 / 0.855	31.12 / 0.872
PromptIR	29.84 / 0.855	31.23 / 0.875
PromptCIR	29.93 / 0.856	31.31 / 0.875
MoE-DiffIR	26.85 / 0.796	26.49 / 0.823
DFPIR	29.59 / 0.850	30.86 / 0.868
Ours (Stage 1)	29.40 / 0.851	30.59 / 0.866
Ours (Stage 2)	**30.24 / 0.863**	**31.72 / 0.882**

Table 3: Quantitative comparison on a codec type in D_2 (WebP, averaged over five QFs), on LIVE1 and DIV2K in terms of PSNR↑/SSIM↑. The best result is in **bold**.

Methods	AVC (QP=42, 37, 32, 27)	
	LIVE1	DIV2K
	PSNR/SSIM	PSNR/SSIM
AirNet	32.29 / 0.887	33.56 / 0.900
PromptIR	32.34 / 0.887	33.64 / 0.902
PromptCIR	32.43 / 0.887	33.74 / 0.902
MoE-DiffIR	27.76 / 0.820	27.53 / 0.847
DFPIR	32.34 / 0.887	33.54 / 0.900
Ours (Stage 1)	32.17 / 0.887	33.31 / 0.898
Ours (Stage 2)	**32.60 / 0.891**	**33.89 / 0.905**

Table 4: Quantitative comparison on an unseen codec type in D_u (AVC, averaged over four QPs), on LIVE1 and DIV2K in terms of PSNR↑/SSIM↑. The best result is in **bold**.

Comparisons

The results are evaluated on the RGB channel using PSNR and SSIM. The compared models include three all-in-one models: AirNet (Li et al. 2022), PromptIR (Potlapalli et al. 2023), DFPIR (Tian et al. 2025), and one CIR model: PromptCIR (Li et al. 2024a), and one UCIR model: MoE-DiffIR (Ren et al. 2024). For a fair comparison, we retrain them using their officially released code. Specifically, each model is first pre-trained on the dataset generated under the D_1 conditions and then fine-tuned with the dataset from the D_1 and D_2 conditions, under identical training parameters, including batch size and number of epochs. All models are evaluated after the full training procedure, except for ours, which is evaluated at both stages, denoted as Ours (Stage 1) and Ours (Stage 2), respectively. The results of the compared models trained only after the first stage are provided in the supplementary material.

Table 1 summarizes average performance under the setting of D_1 and shows that our method achieves higher PSNR and SSIM than all compared models in most cases. In particular, it achieves a PSNR gain of up to 0.95dB compared to PromptCIR (Li et al. 2024a), which adopts a similar encoder-decoder architecture. Notably, Ours (Stage 2) maintains performance on D_1, confirming no degradation on previously encountered configurations.

Tables 2 and 3 evaluate performance under the settings of D_2. Under these settings, which were not seen during the first-stage training, Ours (Stage 1) struggles under these settings, while Ours (Stage 2) achieves the best performance, demonstrating the model's ability to adapt to novel condi-

Methods	JPEG→VVC		VVC→JPEG		VVC→C_{PSNR}		C_{PSNR}→VVC	
	LIVE1	DIV2K	LIVE1	DIV2K	LIVE1	DIV2K	LIVE1	DIV2K
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
AirNet	27.64/0.804	29.27/0.841	28.89/0.824	30.07/0.842	28.94/0.819	30.05/0.839	27.29/0.765	28.89/0.811
PromptIR	27.56/0.799	29.32/0.840	28.79/0.822	30.05/0.842	28.92/0.820	30.06/0.840	27.28/0.765	28.88/0.812
PromptCIR	27.15/0.760	29.44/0.841	28.85/0.822	30.09/0.842	28.94/0.820	30.08/0.840	27.31/0.765	28.91/28.85
MoE-DiffIR	25.84/0.759	25.36/0.783	27.17/0.802	26.63/0.822	26.26/0.803	26.26/0.803	25.22/0.757	25.22/0.757
DFPIR	26.51/0.770	28.03/0.804	28.79/0.821	30.03/0.840	28.95/**0.822**	30.09/**0.842**	27.27/0.767	28.86/0.813
Ours (Stage 1)	27.98/**0.811**	29.71/0.847	**28.99/0.827**	**30.20/0.845**	**28.98**/0.821	30.11/0.841	27.27/**0.768**	28.89/**0.815**
Ours (Stage 2)	**28.02/0.811**	**29.80/0.848**	28.96/0.826	30.17/0.844	**28.98**/0.821	**30.12**/0.841	**27.33**/0.767	**28.93**/0.813

Table 5: Quantitative comparison on unseen double-compressed combinations (first→second compression) in D_u, reported on LIVE1 and DIV2K in terms of PSNR↑/SSIM↑. The best result is in **bold**.

tions through RL-based fine-tuning. Tables 4 and 5 present evaluations under fully unseen scenarios of D_u. Similarly, Ours (Stage 1) shows degraded performance, whereas Ours (Stage 2) outperforms other methods in most cases. Across all these settings, our method demonstrates consistent performance improvements, indicating robustness and generalization ability. These results validate the effectiveness of CCM and RL-based optimization strategy, which together enable our model to handle diverse and unseen compression artifacts. Further evaluations, including PSNR-B (Yim and Bovik 2011) and detailed per-codec and per-level results, are provided in the supplementary material.

Additionally, we provide qualitative comparisons in Fig. 2. Our method shows a high capability for removing diverse compression artifacts. This effectiveness is particularly evident in the D_2 and D_u settings, shown in the third and fourth rows. In these examples, our model is successful at restoring fine details while mitigating severe visual distortions.

Ablation Studies

Codec-aware conditioning. We investigate the effect of each input to CCM. As shown in Table 6, we compare different configurations by selectively enabling the image feature F_r, codec type embedding F_t, and compression level feature F_l when constructing the fused condition representation. The results indicate that using both F_t and F_l provides consistent improvements across both LIVE1 and DIV2K datasets. While the inclusion of either feature alone yields partial benefits, combining the two leads to the best performance. This suggests that codec-aware restoration benefits from not only knowing the compression type but also understanding its relative strength or severity.

RL objective. We also analyze the effect of the RL-based loss used in the second-stage training. Table 7 summarizes the results when training with only the standard reconstruction loss $\mathcal{L}_{rec}$ versus the full objective including the policy and value losses, $\mathcal{L}_{pol}$ and $\mathcal{L}_{val}$. Notably, the first row in the table, where all loss components are marked as inactive, corresponds to Ours (Stage 1), which is trained only on D_1 settings. Incorporating the RL-based losses yields consistent improvements in PSNR and SSIM across both datasets. These results suggest that the reward-based feedback helps the model generalize better across diverse compression artifacts, making it effective for both traditional and learning-based codecs in the UCIR setting.

Conditions			LIVE1		DIV2K	
F_r	F_t	F_l	PSNR	SSIM	PSNR	SSIM
✗	✗	✗	31.07	0.875	32.82	0.896
✓	✗	✗	31.09	0.876	32.86	0.897
✓	✓	✗	31.13	0.877	32.90	0.897
✓	✗	✓	31.12	0.876	32.89	0.897
✓	✓	✓	31.17	0.880	32.95	0.900

Table 6: Ablation study on the impact of each condition. Results are averaged over all D_1 settings on LIVE1 and DIV2K in terms of PSNR↑/SSIM↑.

Objectives		LIVE1		DIV2K	
$\mathcal{L}_{\text{rec}}$	$\mathcal{L}_{pol} + \mathcal{L}_{val}$	PSNR	SSIM	PSNR	SSIM
✗	✗	30.64	0.847	32.03	0.866
✓	✗	31.60	0.857	32.91	0.878
✓	✓	31.75	0.860	33.08	0.881

Table 7: Ablation study on each loss component used in the second stage. Results averaged over all D_2 settings on LIVE1 and DIV2K in terms of PSNR↑/SSIM↑.

Conclusion

In this paper, we present a unified framework for UCIR that handles diverse compression artifacts across both traditional and learned codecs. Our method introduces CCM that explicitly encodes both codec type and compression level, enabling the restoration network to adapt to diverse compression artifacts. We further incorporate an RL-based objective in a two-stage training scheme, which enhances the model's ability to learn from both differentiable and non-differentiable compression processes. To support this task, we construct a UCIR dataset covering multiple codecs and compression levels. Experimental results show that our method performs well across a broad set of compression scenarios, demonstrating its generalizability and effectiveness.

Acknowledgments

This work was supported by the Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2017-II170072, Development of Audio/Video Coding and Light Field Media Fundamental Technologies for Ultra Realistic Tera-media).

References

Agustsson, E.; and Timofte, R. 2017. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 126–135.

Arbeláez, P.; Maire, M.; Fowlkes, C.; and Malik, J. 2011. Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5): 898–916.

Bross, B.; Wang, Y.-K.; Ye, Y.; Liu, S.; Chen, J.; Sullivan, G. J.; and Ohm, J.-R. 2021. Overview of the Versatile Video Coding (VVC) Standard and its Applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10): 3736–3764.

Cao, Q.; Lin, L.; Shi, Y.; Liang, X.; and Li, G. 2017. Attention-Aware Face Hallucination via Deep Reinforcement Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 690–698.

Chen, X.; Wang, X.; Zhou, J.; Qiao, Y.; and Dong, C. 2023. Activating More Pixels in Image Super-Resolution Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 22367–22377.

Cheng, Z.; Sun, H.; Takeuchi, M.; and Katto, J. 2020. Learned Image Compression With Discretized Gaussian Mixture Likelihoods and Attention Modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7939–7948.

Cho, S.; Lee, J.; Kim, J.; Kim, Y.; Kim, D.-W.; Chung, J.; and Jung, S. 2019. Low Bit-rate Image Compression based on Post-processing with Grouped Residual Dense Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 1–4.

Deng, J.; Wang, L.; Pu, S.; and Zhuo, C. 2020. Spatio-Temporal Deformable Convolution for Compressed Video Quality Enhancement. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07): 10696–10703.

Ding, Q.; Shen, L.; Yu, L.; Yang, H.; and Xu, M. 2024. Blind Quality Enhancement for Compressed Video. *IEEE Transactions on Multimedia*, 26: 5782–5794.

Dong, C.; Deng, Y.; Loy, C. C.; and Tang, X. 2015. Compression Artifacts Reduction by a Deep Convolutional Network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 576–584.

Furuta, R.; Inoue, N.; and Yamasaki, T. 2020. PixelRL: Fully Convolutional Network With Reinforcement Learning for Image Processing. *IEEE Transactions on Multimedia*, 22(7): 1704–1719.

Ginesu, G.; Pintus, M.; and Giusto, D. D. 2012. Objective assessment of the WebP image coding algorithm. *Signal Processing: Image Communication*, 27(8): 867–874.

Guo, J.; Chen, Z.; Li, W.; Guo, Y.; and Zhang, Y. 2025. Compression-Aware One-Step Diffusion Model for JPEG Artifact Removal. arXiv:2502.09873.

Hu, Y.; He, H.; Xu, C.; Wang, B.; and Lin, S. 2018. Exposure: A White-Box Photo Post-Processing Framework. *ACM Transactions on Graphics (TOG)*, 37(2): 1–17.

Huang, J.-B.; Singh, A.; and Ahuja, N. 2015. Single Image Super-Resolution From Transformed Self-Exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5197–5206.

Jiang, J.; Zhang, K.; and Timofte, R. 2021. Towards Flexible Blind JPEG Artifacts Removal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 4997–5006.

Kim, Y.; Soh, J. W.; and Cho, N. I. 2020. AGARNet: Adaptively Gated JPEG Compression Artifacts Removal Network for a Wide Range Quality Factor. *IEEE Access*, 8: 20160–20170.

Kim, Y.; Soh, J. W.; Park, J.; Ahn, B.; Lee, H.-S.; Moon, Y.-S.; and Cho, N. I. 2020. A Pseudo-Blind Convolutional Neural Network for the Reduction of Compression Artifacts. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(4): 1121–1135.

Li, B.; Li, X.; Lu, Y.; Feng, R.; Guo, M.; Zhao, S.; Zhang, L.; and Chen, Z. 2024a. PromptCIR: Blind Compressed Image Restoration with Prompt Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 6442–6452.

Li, B.; Liu, X.; Hu, P.; Wu, Z.; Lv, J.; and Peng, X. 2022. All-in-One Image Restoration for Unknown Corruption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 17452–17462.

Li, X.; Li, B.; Jin, Y.; Lan, C.; Zhu, H.; Ren, Y.; and Chen, Z. 2024b. UCIP: A Universal Framework for Compressed Image Super-Resolution Using Dynamic Prompt. In *European Conference on Computer Vision (ECCV)*, 107–125.

Mentzer, F.; Toderici, G. D.; Tschannen, M.; and Agustsson, E. 2020. High-Fidelity Generative Image Compression. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 11913–11924.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 48, 1928–1937.

Ogino, Y.; Toizumi, T.; and Ito, A. 2025. CURVE: CLIP-Utilized Reinforcement Learning for Visual Image Enhancement via Simple Image Processing. arXiv:2505.23102.

Park, J.; Lee, J.-Y.; Yoo, D.; and Kweon, I. S. 2018. Distort-and-Recover: Color Enhancement Using Deep Reinforcement Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5928–5936.

Potlapalli, V.; Zamir, S. W.; Khan, S. H.; and Shahbaz Khan, F. 2023. PromptIR: Prompting for All-in-One Image Restoration. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 71275–71293.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 139, 8748–8763.

Ren, Y.; Li, X.; Li, B.; Wang, X.; Guo, M.; Zhao, S.; Zhang, L.; and Chen, Z. 2024. MoE-DiffIR: Task-Customized Diffusion Priors for Universal Compressed Image Restoration. In *European Conference on Computer Vision (ECCV)*, 116–134.

Sheikh, H. 2005. LIVE image quality assessment database release 2. *http://live. ece. utexas. edu/research/quality*. Accessed: 2025-03-26.

Sullivan, G. J.; Ohm, J.-R.; Han, W.-J.; and Wiegand, T. 2012. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12): 1649–1668.

Tian, X.; Liao, X.; Liu, X.; Li, M.; and Ren, C. 2025. Degradation-Aware Feature Perturbation for All-in-One Image Restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 28165–28175.

Timofte, R.; Agustsson, E.; Van Gool, L.; Yang, M.-H.; and Zhang, L. 2017. NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 114–125.

Wallace, G. K. 1991. The JPEG still picture compression standard. *Communications of the ACM*, 34(4): 30–44.

Wang, M.; Fu, X.; Sun, Z.; and Zha, Z.-J. 2020. JPEG Artifacts Removal via Compression Quality Ranker-Guided Networks. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, 566–572. Main track.

Wiegand, T.; Sullivan, G.; Bjontegaard, G.; and Luthra, A. 2003. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7): 560–576.

Yang, Z.; Chen, H.; Qian, Z.; Yi, Y.; Zhang, H.; Zhao, D.; Wei, B.; and Xu, Y. 2024. All-In-One Medical Image Restoration via Task-Adaptive Routing. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 67–77.

Yim, C.; and Bovik, A. C. 2011. Quality Assessment of Deblocked Images. *IEEE Transactions on Image Processing*, 20(1): 88–98.

Yu, K.; Dong, C.; Lin, L.; and Loy, C. C. 2018. Crafting a Toolchain for Image Restoration by Deep Reinforcement Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2443–2452.

Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; and Finn, C. 2020. Gradient Surgery for Multi-Task Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 5824–5836.

Zeyde, R.; Elad, M.; and Protter, M. 2012. On Single Image Scale-Up Using Sparse-Representations. In *Curves and Surfaces*, 711–730.

Zhang, H.; Jung, C.; Zou, D.; and Li, M. 2023. WCDANN: A Lightweight CNN Post-Processing Filter for VVC-Based Video Compression. *IEEE Access*, 11: 83400–83413.

Zhang, R.; Guo, L.; Huang, S.; and Wen, B. 2021. ReLLIE: Deep Reinforcement Learning for Customized Low-Light Image Enhancement. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, 2429–2437.

Zhu, J.; Zhu, X.; Wang, W.; Wang, X.; Li, H.; Wang, X.; and Dai, J. 2022. Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2664–2678.