Task-Oriented JSCC with Adaptive Deep Compressed Sensing

Mohammad Amin Jarrahi, *Student Member, IEEE,* Eirina Bourtsoulatze, *Member, IEEE,* and Vahid Abolghasemi, *Senior Member, IEEE*

***Abstract*—In this letter, we propose a Task-oriented Joint Source-Channel Coding framework based on Compressed Sensing (TCS-JSCC), designed to enable end-to-end wireless image transmission for downstream tasks without requiring full image reconstruction. The framework integrates a rate-adaptive CS module with the unique ability to adaptively encode semantically meaningful features informed by both image contents and channel conditions. Specifically, semantic feature representations and signal-to-noise ratio (SNR) are jointly exploited to enable dynamic rate allocation and enhance robustness against channel impairments. A lightweight decoder then processes the received features to perform task-specific inference, such as face detection and image classification, directly on the compressed data. Experimental results demonstrate that TCS-JSCC achieves competitive task accuracy under varying channel conditions, offering a scalable and effective solution for edge-intelligent, bandwidth-constrained applications.**

***Index Terms*—Joint source-channel coding, task-oriented communication, compressed sensing.**

I. Introduction

JOINT source-channel coding (JSCC) has emerged as a promising solution for semantic communication. Well-established frameworks, including DeepJSCC [1] and more recent extensions such as SwinJSCC [2], outperform traditional separate source-channel coding (SSCC) by jointly optimizing compression and transmission for image reconstruction. These models are primarily designed to maximize reconstruction fidelity, and hence, are not inherently optimized for downstream tasks (e.g. classification or detection), leading to unnecessary computational overhead and bandwidth usage. Moreover, convolutional neural network (CNNs) and attention-based JSCC methods typically treat all image regions equally, overlooking task-critical features, while adaptive schemes such as ADJSCC [3] employ complex channel integration modules that are unsuitable for lightweight edge deployment.

Recent advances in semantic JSCC have shifted focus from image reconstruction toward optimizing downstream task performance, including classification, detection, and segmentation [4]–[6]. For example, Wu et al. [7] proposed a method that transmits only segmentation features, and feature embeddings are used for stereo-vision 3D object detection [8]. These works highlight the potential of task-specific communication but often employ fixed or globally adaptive compression strategies and do not fully exploit fine-grained content importance or channel state information [9]. This limits their ability to efficiently balance semantic relevance and transmission robustness under dynamic wireless conditions.

The authors are with the School of Computer Science and Electronic Engineering (CSEE), University of Essex, Colchester, United Kingdom (emails: m.jarrahi, e.bourtsoulatze and v.abolghasemi@essex.ac.uk).

In this work, we present TCS-JSCC, a task-oriented JSCC framework with adaptive compressed sensing (CS). TCS-JSCC learns to compress, transmit, and decode only task-relevant features, informed by the channel conditions. A rate-adaptive CS module, guided by a sampling rate prediction network (SRNet), dynamically adjusts block-wise sampling rates based on both content importance and channel SNR. To improve robustness, we further incorporate lightweight SNR-aware feature modulation via Feature-wise Linear Modulation (FiLM) [10]. The decoder directly produces task-specific outputs from received features, avoiding reconstruction overhead. Recent importance-aware and channel-adaptive semantic communication approaches highlight allocating more bits to task-critical content under varying SNRs [11]. Our design differs by coupling block-wise rate selection with task-driven training and lightweight SNR conditioning, improving task accuracy while avoiding heavy attention modules. While general-purpose, we demonstrate the framework on face detection (i.e., automatically locating the presence of human faces) and image classification as two representative tasks. Key contributions of this letter are:

- We introduce a task-oriented JSCC pipeline that integrates block-wise CS into end-to-end training, transmitting only task-relevant features without image reconstruction.
- We design a rate-adaptive CS scheme that enables block-wise sampling guided by both semantic importance and channel SNR through SRNet.
- We integrate SNR-aware FiLM modulation for efficient channel adaptation, making the framework practical for downstream tasks directly.

While our prior works [12]–[14] addressed CS and JSCC primarily from a signal reconstruction perspective, they laid the foundation for the present design. Here, we move beyond reconstruction to explicitly target task performance, introducing TCS-JSCC framework that integrates block-level rate adaptivity with lightweight channel conditioning in a unified task-oriented end-to-end architecture.

II. System Model

Fig. 1 illustrates the proposed framework where an input RGB image $\boldsymbol{x} \in \mathbb{R}^{H\times W\times 3}$, is compressed using content- and SNR-adaptive rates, and transmitted over a wireless channel. The channel is modeled as a block-fading Additive White Gaussian Noise (AWGN) channel:

$$\hat{\boldsymbol{z}} = h\boldsymbol{z} + \boldsymbol{n} \tag{1}$$

where $\boldsymbol{z}$ is the channel input, $\hat{\boldsymbol{z}}$ the channel output and $\boldsymbol{n} \sim \mathcal{CN}(0, \sigma_n^2 \boldsymbol{I})$ is a vector of independent identically distributed (i.i.d.) noise samples drawn from a complex Gaussian

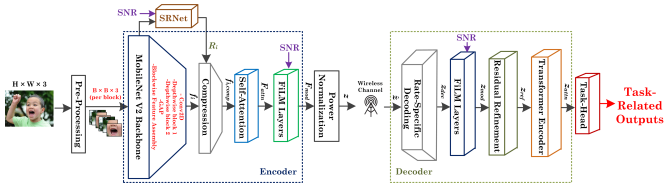


Fig. 1. Block diagram of the proposed TCS-JSCC framework and architecture design of the encoder and decoder networks.

distribution, with σ_n^2 being the noise variance. The fading channel coefficient $h \sim \mathcal{CN}(0,1)$ is a complex random variable with a Rayleigh-distributed amplitude and a uniformly distributed phase. We define the signal-to-noise ratio as $\text{SNR} = 10\log_{10}\frac{\mathbb{E}[|h|^2]\cdot P}{\sigma_n^2}$ where $P = 1$ is the average transmitted power, and the fading coefficient h is constant per image but varies across transmissions. The estimated channel SNR at the receiver is fed back for dynamic rate adaptation. At the receiver, decoded task-specific features are aggregated to execute the task.

III. TCS-JSCC

A. Encoder Architecture

The encoder (Fig. 1), consists of five main components: preprocessing, feature extraction, rate-adaptive compressed sensing, attention-based encoding, and SNR-aware feature adaptation. During the preprocessing, the input image $\boldsymbol{x}$ is globally normalized to the range [0, 1] and then divided into non-overlapping blocks of size $B \times B$. The block size B balances feature granularity and computational cost. Smaller blocks better preserve localized details but increase processing overhead due to larger block count, while larger blocks reduce computational overhead yet may blend heterogeneous regions. We set $B = 16$ as a practical trade-off.

Following the preprocessing step, a lightweight modified version of MobileNetV2 [15] extracts features from each block. We denote 2D convolutional layers as Conv2D (c_{in}, c_{out}, k) and fully connected layers as FC (c_{in}, c_{out}), where c_{in} and c_{out} are the number of input and output channels, respectively, and k is the kernel size. The architecture of the feature extraction network comprises an intital convolution block, two depthwise blocks and a feature extraction block, and is as follows: Conv2D (3, 32, 3) $\rightarrow$ BatchNorm $\rightarrow$ ReLU6 $\rightarrow$ depthwise Conv2D (32, 32, 3) $\rightarrow$ BatchNorm $\rightarrow$ ReLU6 $\rightarrow$ pointwise Conv2D (32, 64, 1) $\rightarrow$ BatchNorm $\rightarrow$ pointwise Conv2D (64, 384, 1) $\rightarrow$ BatchNorm $\rightarrow$ ReLU6 $\rightarrow$ depthwise Conv2D (384, 384, 3) $\rightarrow$ BatchNorm $\rightarrow$ ReLU6$\rightarrow$ pointwise Conv2D (384, 128, 1) $\rightarrow$ BatchNorm $\rightarrow$ GlobAvgP $\rightarrow$ Conv2D (128, 256, 1) $\rightarrow$ ReLU6. We denote the extracted features for the i-th block as $\boldsymbol{f}_i \in \mathbb{R}^{256\times 1}$.

To adaptively compress image regions, we propose a multi-rate CS scheme that dynamically selects sampling rates for each feature vector $\boldsymbol{f}_i$ from a discrete set $\mathcal{R}$ based on task-related feature importance and channel conditions. A lightweight module, termed SRNet, predicts the optimal sampling rate $R_i \in \mathcal{R}$ for the i-th feature, controlling the compression strength. SRNet consists of: FC(257, 128) $\rightarrow$ ReLU$\rightarrow$ FC(128, $|\mathcal{R}|$) $\rightarrow$ Softmax, where $|\mathcal{R}|$ is the number of available sampling rates. SRNet takes $\boldsymbol{f}_i$ and the channel SNR as input, and outputs a probability distribution $\boldsymbol{r}_i$ across the available rates in $\mathcal{R}$, with R_i selected as the most probable rate.

Unlike fixed or global rate allocation strategies, which suboptimally allocate similar rate budgets to background and salient regions [16], SRNet learns to allocate higher sampling rates to edge-rich blocks at low SNR and lower sampling rates elsewhere, improving task accuracy for the same bandwidth. SRNet amortizes rate selection via a lightweight two-layer network, achieving comparable adaptivity with far lower complexity than costly policy-network controllers.

Given the predicted sampling rate R_i, each feature is compressed using learned projection matrices that map high-dimensional feature vectors into a lower-dimensional space:

$$\boldsymbol{f}_{i,\text{comp}} = \Phi_{R_i}\cdot\boldsymbol{f}_i + \boldsymbol{b}_{R_i}, \quad \Phi_{R_i} \in \mathbb{R}^{\lfloor R_i\cdot 256\rfloor\times 256} \tag{2}$$

where $\boldsymbol{f}_{i,\text{comp}}$ is the compressed feature vector for the i-th block, and $\boldsymbol{b}_{R_i}$ and Φ_{R_i} are rate-specific bias vector and projection matrix, respectively, for rate $R_i \in \mathcal{R}$. The rate-specific projection matrices and biases are learned jointly with all other network components during the end-to-end training.

The compressed feature vectors $\boldsymbol{f}_{i,\text{comp}}$ are then aggregated into a block-structured tensor $\boldsymbol{F}_{\text{comp}}$ and refined using a self-attention mechanism. Specifically, $\boldsymbol{F}_{\text{comp}}$ is projected via 1D convolutional layers to obtain Query (Q), Key (K), and Value (V) representations, and standard scaled dot-product attention is applied:

$$\boldsymbol{F}_{\text{attn}} = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

where d_k denotes the key dimension. This attention operates across spatial blocks to model long-range dependencies and prioritize informative features, with a residual connection used to improve gradient flow.

To improve robustness to channel noise, the feature vector $\boldsymbol{F}_{\text{attn}}$ is processed using an SNR-aware FiLM-based encoder. The scalar SNR is passed through two FC layers with ReLU activations to generate the FiLM parameters $\gamma_{\text{enc}}(\text{SNR})$ and $\beta_{\text{enc}}(\text{SNR})$, which provide channel-wise scaling and bias, respectively [10]. This transforms the scalar SNR into feature-level modulation parameters, enabling adaptive encoding under varying channel conditions. The resulting modulation is:

$$\boldsymbol{F}_{\text{mod}} = \gamma_{\text{enc}}(\text{SNR})\odot\boldsymbol{F}_{\text{attn}} + \beta_{\text{enc}}(\text{SNR}) \tag{4}$$

where $\odot$ denotes element-wise multiplication. FiLM applies lightweight channel-wise affine transformations, offering effective SNR adaptation without the computational overhead of attention mechanisms.

To meet the wireless power transmission constraints, the feature power is constrained via the power normalization rule described in [1]. The final encoded representation $\boldsymbol{z}$ is transmitted over the wireless channel.

B. Decoder Architecture

At the decoder (Fig. 1) the received noisy compressed features in $\hat{\boldsymbol{z}}$ are first reconstructed to their original dimensions

(256×1) via inverse projections in a rate-specific manner. Specifically, for each projection matrix Φ_{R_i} at the encoder, a corresponding inverse projection matrix is learned at the decoder, to reconstruct features compressed at rate R_i. This operation inverts the encoder's compression step and yields the decompressed feature tensor $\boldsymbol{z}_{\text{dec}}$.

Next, to mitigate channel-induced noise, we apply a channel-adaptive FiLM layer conditioned on SNR. This layer performs an affine transformation on the decoded features:

$$\boldsymbol{z}_{\text{mod}} = \gamma_{\text{dec}}(\text{SNR}) \odot \boldsymbol{z}_{\text{dec}} + \beta_{\text{dec}}(\text{SNR}) \tag{5}$$

where $\gamma_{\text{dec}}(\text{SNR})$ and $\beta_{\text{dec}}(\text{SNR})$ are channel-wise scaling and shifting parameters generated from the SNR value via two fully-connected layers. This step produces the channel-modulated feature map $\boldsymbol{z}_{\text{mod}}$.

Following feature modulation, we apply a residual refinement block to suppress local distortions in $\boldsymbol{z}_{\text{mod}}$, mainly block-boundary inconsistencies and high-frequency perturbations introduced by rate-dependent inverse projection and channel noise, before the Transformer module. The refinement block consists of a residual branch set up as: Conv2D (256,128,3) $\rightarrow$ BatchNorm $\rightarrow$ LeakyReLU $\rightarrow$ Conv2D (128,128,3), and a skip connection with a Conv2D (256,128,1) layer, used to match channel dimensions and preserve stable gradient flow. We adopt two convolutional layers in the residual branch as the smallest configuration that provides a non-trivial local correction with limited decoder complexity; deeper stacks offer diminishing returns at the cost of higher parameter count, while removing the refinement block degrades performance, particularly at low SNR. To model global dependencies across all feature blocks, the refined feature map $\boldsymbol{z}_{\text{ref}}$ is then processed by a standard Transformer encoder module, producing an attention-aware feature map $\boldsymbol{z}_{\text{attn}}$. This self-attention mechanism allows the network to capture long-range spatial relationships. Finally, the Transformer block is followed by a lightweight task specific network which processes $\boldsymbol{z}_{\text{attn}}$ in a way that is appropriate for the considered task.

C. Loss Function

The proposed TCS-JSCC model is optimized end-to-end using a multi-objective loss function $\mathcal{L}_{\text{total}}$, designed to jointly enhance task accuracy, preserve semantic feature information, and ensure efficient rate adaptation:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{feat}} + \lambda_{\text{rate}}\mathcal{L}_{\text{rate}} \tag{6}$$

where λ_{rate} is a scalar weight that balances the contribution of the $\mathcal{L}_{\text{rate}}$ component.

Task-specific loss $\mathcal{L}_{\text{task}}$ is the primary loss for the task-related objectives. For example, for a classification task, this can take the form of a categorical cross entropy. To maintain semantic consistency across the end-to-end communication process, we introduce a feature alignment loss $\mathcal{L}_{\text{feat}}$ between the encoder's semantic features $\boldsymbol{f}_i$ and the decoder's reconstructed representation $\boldsymbol{z}_{\text{dec},i}$, where $\boldsymbol{z}_{\text{dec},i}$ is the decompressed feature for the i-th block. To ensure scale-invariant comparisons and reduce the impact of stochastic channel noise, we ℓ_2-normalize both vectors and define the feature consistency loss as:

$$\mathcal{L}_{\text{feat}} = \frac{1}{N}\sum_{i=1}^{N}\left[\lambda_{\text{mse}}\|\boldsymbol{f}_i - \boldsymbol{z}_{\text{dec},i}\|_2^2 + \lambda_{\cos}\left(1 - \frac{\boldsymbol{f}_i \cdot \boldsymbol{z}_{\text{dec},i}}{\|\boldsymbol{f}_i\|\|\boldsymbol{z}_{\text{dec},i}\|}\right)\right] \tag{7}$$

where N is the number of blocks. The first term preserves numerical consistency via Mean Squared Error (MSE), while the second promotes angular alignment via cosine similarity. The weights λ_{mse} and $\lambda_{\cos}$ balance the two objectives. This formulation encourages the decoder to reconstruct meaningful semantic features despite compression and channel distortion.

To train SRNet for adaptive rate allocation, we utilize a self-regularized formulation that encourages the network to assign sampling rates based on both feature importance and channel conditions, without requiring explicit rate labels. We define a rate regularization loss that penalizes unnecessary bandwidth usage while preserving task-critical information as:

$$\mathcal{L}_{\text{rate}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{R\in\mathcal{R}} \boldsymbol{r}_i(R)\, R \tag{8}$$

where $\boldsymbol{r}_i(R)$ is the predicted distribution over $\mathcal{R}$ and $\sum_{R\in\mathcal{R}} \boldsymbol{r}_i(R)\, R$ is the expected sampling rate for the i-th block under $\boldsymbol{r}_i(R)$. The loss term $\mathcal{L}_{\text{rate}}$ is differentiable with respect to $\boldsymbol{r}_i(R)$ and allows gradient backpropagation during training.

IV. Results and Discussion

A. Face Detection Task

To process $\boldsymbol{z}_{\text{attn}}$, we first employ a multi-scale fusion module with three parallel convolutional layers with kernel sizes 1×1, 3×3, and 5×5 to extract features at multiple receptive fields and create a rich, multi-scale feature representation to handle faces at various scales. This representation is then fed into two parallel prediction heads: a regression head that predicts the four bounding box coordinates for each location, and a classification head with a sigmoid activation that predicts the corresponding confidence score.

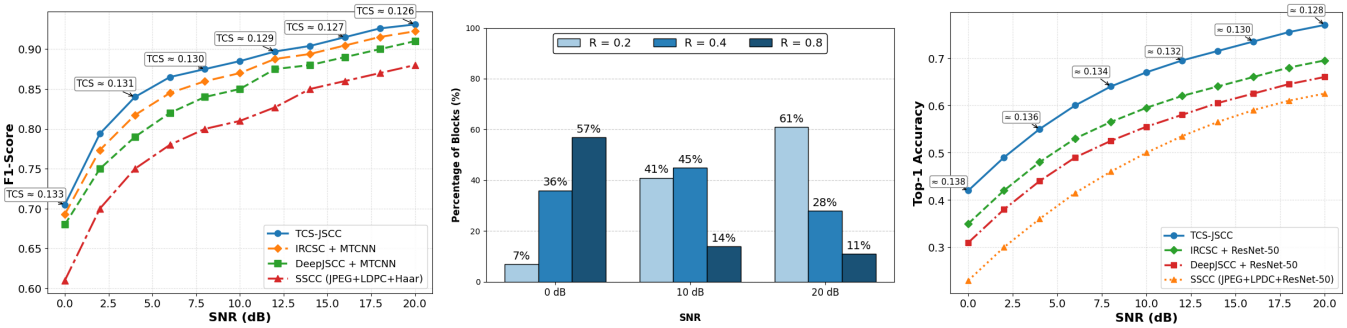
The task-specific term $\mathcal{L}_{\text{task}}$ of the loss function in Eq. (6) ensures accurate classification and localization of faces and is defined as:

$$\mathcal{L}_{\text{task}} = \mathcal{L}_{\text{cls}} + \lambda_{\text{IoU}}\mathcal{L}_{\text{IoU}} \tag{9}$$

where $\mathcal{L}_{\text{cls}}$ is a classification loss and $\mathcal{L}_{\text{IoU}}$ is a bounding box regression loss. The classification loss uses binary cross-entropy to distinguish face regions from the background. For each prediction $\hat{y}_i$ and ground truth label $y_i \in \{0,1\}$, it is defined as:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{M}\sum_{i=1}^{M}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \tag{10}$$

where M is the total number of classification predictions used to compute the loss. For localization, we use the Generalized IoU (GIoU) loss, which remains informative even when predicted and ground-truth boxes do not overlap by penalizing the normalized area of their smallest enclosing box. The localization loss is applied only to positive samples. In all experiments,



(a) (b) (c)

Fig. 2. Comprehensive performance evaluation of the TCS-JSCC framework across varying channel SNRs. (a) F1-score (y-axis), and bandwidth compression (boxed numbers) against varying SNR values [0-20] dB for different methods. (b) The block-wise SRNet rate allocation adapts to channel conditions, minimizing bandwidth at high SNRs. (c) Top-1 accuracy for image classification compared to digital and analog baselines.

TABLE I
COMPONENT ABLATION STUDY ISOLATING SRNET AND FILM ACROSS REPRESENTATIVE SNR VALUES.

Variant	SNR = 0 dB				SNR = 10 dB				SNR = 20 dB			
	Prec.	Rec.	F1	BW ratio	Prec.	Rec.	F1	BW ratio	Prec.	Rec.	F1	BW ratio
Full (SRNet+FiLM)	**0.760**	**0.658**	**0.705**	**0.133**	**0.915**	**0.856**	**0.885**	**0.130**	**0.953**	**0.910**	**0.931**	**0.126**
w/o FiLM	0.748	0.642	0.691	0.133	0.904	0.842	0.872	0.130	0.945	0.900	0.922	0.126
w/o SRNet (fixed sampling rate)	0.735	0.628	0.677	0.134	0.891	0.829	0.859	0.131	0.937	0.892	0.914	0.127
w/o SRNet & w/o FiLM	0.721	0.612	0.661	0.134	0.879	0.816	0.846	0.131	0.928	0.880	0.903	0.127

the loss weights are fixed to $\lambda_{\text{rate}} = 0.1$, $\lambda_{\text{mse}} = \lambda_{\cos} = 0.5$, and $\lambda_{\text{IoU}} = 1.0$. A sensitivity analysis around these defaults shows that moderate variations do not cause instability or significant performance changes. Specifically, λ_{rate} mainly affects the learned bandwidth allocation, while λ_{mse} and $\lambda_{\cos}$ primarily influence robustness at low SNR, indicating that the framework is not overly sensitive to precise tuning.

The proposed TCS-JSCC model is evaluated on the WIDER Face dataset [17] under simulated wireless channels with SNR values from 0 to 20 dB. Unlike conventional schemes trained at a fixed SNR, we adopt a curriculum learning strategy: training begins at high SNR (20 dB) and gradually introduces lower SNRs until 0 dB, which stabilizes convergence and improves robustness across conditions. The model is trained using Adam optimizer with learning rate 10^{-4}, batch size 32, and weight decay 10^{-5}. During inference, the SRNet dynamically selects block-wise sampling rates $R_i \in \{0.2, 0.4, 0.8\}$ based on feature importance and channel SNR, enabling adaptive and task-oriented compression.

We benchmark TCS-JSCC against three baselines: (1) traditional SSCC (JPEG, rate-2/3 LDPC, 16-QAM) with Haar Cascade [18]; (2) DeepJSCC [1] + MTCNN [19]; and (3) an importance-aware scheme based on IRCSC [11] that transmits only top features via an SNR-dependent rate policy, using a fine-tuned MTCNN detector. DeepJSCC is trained end-to-end across SNRs using Adam (learning rate=10^{-4}, batch size=32). MTCNN thresholds are 0.9 (boxes) and 0.7 (landmarks).

1) Performance Analysis: TCS-JSCC significantly outperforms all baselines across all SNRs (Fig. 2a), with F1-score rising from 0.705 at 0 dB to 0.931 at 20 dB. This robust performance stems from its adaptive compression strategy that prioritizes task-relevant features under bandwidth constraints. In contrast, DeepJSCC+MTCNN and SSCC lag behind, highlighting the limitations of full-image reconstruction under channel noise. IRCSC+MTCNN improves over DeepJSCC+MTCNN but does not attain the performance of TCS-JSCC, demonstrating the additional gain achieved by our block-wise rate allocation and SNR-conditioned modulation.

Fig. 2a also reports the effective bandwidth compression ratio for TCS-JSCC (channel uses per image divided by $H \times W \times 3$ source symbols). SSCC and DeepJSCC+MTCNN use a fixed ratio of $1/6$ (≈ 0.166). The IRCSC+MTCNN baseline employs an SNR-dependent importance-aware rate control policy; for fair comparison, its average bandwidth ratio is matched to the one of TCS-JSCC. Fig. 2a shows that TCS-JSCC consistently achieves superior performance with lower bandwidth across all SNRs, due to its rate-adaptive CS strategy that samples non-informative blocks (e.g., background) at lower rates while concentrating rate on salient blocks. The bandwidth compression ratio decreases as SNR increases, confirming that the SNR- and rate-adaptive sampling learns to compress task-relevant features more aggressively and add less redundancy under better channel conditions.

In Fig. 2b we report the empirical distribution of SRNet decisions, that is the number of blocks sampled at each rate divided by the total number of blocks at representative SNR values. The observed trend is consistent with the intended SNR-adaptivity: at low SNR the model assigns a larger fraction of blocks to the low-compression branch ($R = 0.8$) to increase feature robustness to severe noise, while at higher SNR it shifts mass toward higher compression ($R = 0.2$) because less redundancy is needed. Concretely, at 0 dB, 57% of blocks are encoded at $R = 0.8$ and only 7% at $R = 0.2$; at 20 dB, 61% of blocks use $R = 0.2$ and 11% use $R = 0.8$. The

medium branch ($R = 0.4$) peaks in the mid-SNR regime (45% at 10 dB), which reflects the transition between robustness-dominant allocation and compression-dominant allocation.

2) Component Ablation Study: To isolate the contribution of SRNet and FiLM modules, we conduct an ablation study where each component is removed while keeping the backbone, training schedule, dataset split, channel model, and evaluation protocol unchanged. Specifically, we compare: (i) the full model, (ii) the model without (w/o) FiLM (FiLM layers replaced by identity mappings), (iii) the model w/o SRNet (all blocks are sampled at a fixed sampling rate), and (iv) w/o SRNet and w/o FiLM. For fairness, we report precision, recall, and F1-score along with the effective bandwidth ratio for each variant. The results, presented in Table I, show consistent performance drops when removing either module, with the largest degradation when both are removed, confirming that SRNet (spatial rate allocation) and FiLM (SNR-conditioned modulation) provide complementary gains.

B. Image Classification Task (ImageNet-1K)

To provide an additional validation, we implement large-scale image classification on ImageNet-1K. In this task, the backbone network remains unchanged. The task module is replaced with a lightweight classifier head. Concretely, we apply global average pooling (GAP) to $\boldsymbol{z}_{\text{attn}}$ to obtain a global feature vector $\boldsymbol{g}$, followed by a linear layer that produces 1000-way logits:

$$\boldsymbol{g} = \mathrm{GAP}(\boldsymbol{z}_{\text{attn}}), \qquad \hat{\boldsymbol{y}} = \mathrm{Softmax}\big(\mathrm{FC}(\boldsymbol{g})\big) \tag{11}$$

The task specific loss term is defined as:

$$\mathcal{L}_{\text{task}} = \mathcal{L}_{\text{cls}} + \lambda_{\text{distill}}\mathcal{L}_{\text{distill}} \tag{12}$$

where $\lambda_{\text{distill}} = 0.5$ and for $\mathcal{L}_{\text{cls}}$, we use the standard cross-entropy loss on ImageNet-1K labels:

$$\mathcal{L}_{\text{cls}} = -\sum_{c=1}^{1000} y_c \log(\hat{y}_c) \tag{13}$$

To strengthen semantic supervision without image reconstruction, we additionally use label-space distillation from a fixed pretrained ImageNet classifier (ResNet-50). For each clean input image, the teacher produces a soft class distribution $\boldsymbol{y}^{\mathrm{T}}$, and we add a Kullback–Leibler (KL) divergence term:

$$\mathcal{L}_{\text{distill}} = \mathrm{KL}\big(\boldsymbol{y}^{\mathrm{T}} \,\|\, \hat{\boldsymbol{y}}\big) \tag{14}$$

The teacher is frozen and is used only during training. At inference, the receiver uses only the GAP+linear head.

Fig. 2c reports Top-1 accuracy versus SNR on the ImageNet-1K validation set under the same channel model and SNR rage as for the face detection task. The curves show a consistent accuracy improvement for TCS-JSCC across all SNR values, with the largest gains in the low-to-moderate SNR regime, which is where robustness to channel noise is most critical. For a fair comparison, the same pretrained ResNet-50 classifier is used as the receiver-side classifier for SSCC, DeepJSCC, and IRCSC baselines, while for TCS-JSCC it is used only as a frozen teacher during training.

V. Conclusion and Future Works

This letter presents a task-oriented JSCC framework that couples rate-adaptive compressed sensing with SNR-aware encoding to directly optimize task performance in the feature domain. The framework is evaluated on two distinct tasks (face detection and classification) and the results demonstrate that TCS-JSCC consistently outperforms prior work, confirming its inherent task-agnostic nature. Future work will extend the framework to other vision tasks, such as semantic segmentation, to further demonstrate its versatility.

References

[1] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, Sept. 2019.

[2] K. Yang, S. Wang, J. Dai, X. Qin, K. Niu, and P. Zhang, "SwinJSCC: Taming swin transformer for deep joint source-channel coding," *IEEE Trans. Cogn. Commun. Netw.*, vol. 11, no. 1, pp. 90–104, Feb. 2024.

[3] J. Xu, B. Ai, W. Chen, A. Yang, P. Sun, and M. Rodrigues, "Wireless image transmission using deep source channel coding with attention modules," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2315–2328, Apr. 2022.

[4] J. Guo, H. Yin, B. Song, Y. Chi, Z. Zhang, C. Yuen, and D. Niyato, "Multi-scale semantic communication for object detection: Single and cross-domain scenarios," *IEEE Trans. Wireless Commun.*, vol. 24, no. 7, pp. 6195–6210, July 2025.

[5] Z. Lyu, G. Zhu, J. Xu, B. Ai, and S. Cui, "Semantic communications for image recovery and classification via deep joint source and channel coding," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 8388–8404, Aug. 2024.

[6] B. Wang, R. Gu, W. Xu, F. Jiang, M. Li, and S. Wang, "Channel-Aware Deep Joint Source-Channel Coding for Multi-Task Oriented Semantic Communication," *IEEE Wireless Commun. Lett.*, vol. 14, no. 5, pp. 1521–1525, May 2025.

[7] J. Wu, C. Wu, Y. Lin, T. Yoshinaga, L. Zhong, X. Chen, and Y. Ji, "Semantic segmentation-based semantic communication system for image transmission," *Digital Communications and Networks*, vol. 10, no. 3, pp. 519–527, June 2024.

[8] Z. Cao, H. Zhang, L. Liang, H. Wang, S. Jin, and G. Y. Li, "Task-Oriented Semantic Communication for Stereo-Vision 3D Object Detection," *IEEE Trans. Commun.*, vol. 73, no. 9, pp. 7552–7567, Sept. 2025.

[9] Q. Zhou, R. Li, Z. Zhao, Y. Xiao, and H. Zhang, "Adaptive bit rate control in semantic communication with incremental knowledge-based HARQ," *IEEE Open Journal of Comms. Soc.*, vol. 3, pp. 1076–1089, 2022.

[10] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proc. 32nd AAAI Conf. on AI*, New Orleans, Louisiana, USA, 2018.

[11] Z. Sun, S. Ma, and S. Li, "Task-Oriented Semantic Communication with Importance-Aware Rate Control," *IEEE Comms Letters*, vol. 29, no. 7, pp. 1520–1524, 2025.

[12] M. A. Jarrahi, E. Bourtsoulatze, and V. Abolghasemi, "Joint source-channel coding for wireless image transmission: A deep compressed-sensing based method," in *Proc. IEEE WCNC*, 2024, pp. 1–6.

[13] ——, "DCS-JSCC: Leveraging deep compressed sensing into JSCC for wireless image transmission," in *Proc. IEEE SPAWC*, 2024, pp. 96–100.

[14] ——, "Rate-Adaptive Joint Source Channel Coding Using Deep Block-Based Compressed Sensing," in *Proc. IEEE MMSP*, 2024, pp. 1–6.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF CVPR*, 2018, pp. 4510–4520.

[16] M. Yang and H.-S. Kim, "Deep joint source-channel coding for wireless image transmission with adaptive rate control," in *Proc. IEEE ICASSP*, 2022, pp. 5193–5197.

[17] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," in *Proc. IEEE/CVF CVPR*, 2016, pp. 5525–5533.

[18] M. Khan, S. Chakraborty, R. Astya, and S. Khepra, "Face detection and recognition using OpenCV," in *Proc. IEEE ICCCIS*, 2019, pp. 116–119.

[19] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, 2016.