Risk-Calibrated Semantic Transmission for Communication-Efficient Heterogeneous Collaborative Inference

Anonymous Authors[1]

Abstract

This paper proposes a risk-calibrated heterogeneous collaborative inference framework that deploys a lightweight CNN at the edge and a high-capacity ViT at the server. The proposed method uses Conformal Risk Control (CRC) to calibrate the edge-side acceptance threshold with a finite-sample guarantee on the risk of incorrect local acceptance, while Adaptive Prediction Sets (APS) and Grad-CAM are combined to generate class-aware saliency maps for selective patch transmission. By transmitting only semantically informative patches when server inference is required, the proposed framework reduces communication cost while preserving inference accuracy. Experimental results on ImageNet demonstrate that the proposed method achieves 81.05% Top-1 accuracy with a communication cost of 0.26, corresponding to a 74% reduction over full-image transmission.

1. Introduction

Wireless communication systems are evolving toward AI-native infrastructures that support intelligent services within the network (Wu et al., 2021; Jung, 2024). Emerging applications such as real-time perception and interactive AI require timely and accurate inference near the network edge (Zhou et al., 2019; Chen & Ran, 2019). However, executing high-capacity neural networks entirely on edge devices remains challenging due to limited on-device computation resources. To address this issue, collaborative inference has emerged as a promising deployment strategy in which edge devices and remote servers jointly execute inference (Kang et al., 2017; Teerapittayanon et al., 2017; Ren et al., 2023).

A representative approach to collaborative inference is split inference, where the early layers of a neural network are executed at the edge and intermediate features are transmitted to the server (Kang et al., 2017; Li et al., 2018). However, the communication benefit of split inference is limited when the intermediate representation is not sufficiently compact. This issue is pronounced in Vision Transformers (ViTs), whose token representations often maintain comparable dimensionality across transformer layers. Therefore, recent collaborative inference methods have shifted toward patch-level transmission, where only task-relevant image patches are offloaded to the server (Dosovitskiy et al., 2021).

Recent attention-aware collaborative inference methods address this issue by transmitting only a subset of image patches selected from edge-side attention scores. Despite these advances, these methods still rely on heuristic confidence thresholds for offloading decisions (Im et al., 2024). In addition, their patch selection is typically tied to class-token attention from an edge ViT, which restricts the framework to homogeneous ViT-to-ViT architectures and may fail when the edge model is uncertain or incorrect (Im et al., 2024; Touvron et al., 2021).

To address these limitations, we propose a communication-efficient and risk-calibrated framework for heterogeneous collaborative inference. The proposed framework deploys a lightweight CNN at the edge and a high-capacity ViT at the server. To enable theoretically grounded risk control without additional training or large-scale auxiliary data, we use Conformal Risk Control (CRC) to calibrate the edge-side decision threshold with statistical risk guarantees (Angelopoulos et al., 2024). For patch transmission, we introduce a class-aware saliency score that combines APS with Grad-CAM (Romano et al., 2020; Selvaraju et al., 2017). It enables the edge device to transmit only informative image patches to the server. Experimental results show that the proposed framework achieves a 74% reduction in communication cost while keeping the accuracy loss within 1 pp compared to the server-only model.

The rest of the paper is organized as follows. Section 2 reviews conformal methods and Grad-CAM. Section 3 presents the proposed framework, including CRC-based edge decision and APS-guided saliency-based patch transmission. Section 4 reports the experimental setup and results. Section 5 concludes the paper.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. Background

2.1. Conformal Prediction and Risk Control

Conformal prediction is a distribution-free uncertainty quantification framework that provides marginal coverage guarantees for arbitrary underlying prediction models (Romano et al., 2020). This coverage guarantee relies only on the exchangeability of the calibration and test samples. Therefore, conformal prediction can be applied post hoc to a pre-trained model without modifying its architecture or retraining its parameters. This property makes conformal methods particularly suitable for deployed edge inference systems, where reliability guarantees must be incorporated without modifying the pre-trained model.

In this work, we use two variants of conformal methods: Adaptive Prediction Sets (APS) and Conformal Risk Control (CRC). APS is used to construct a candidate class set for saliency generation, while CRC is used to calibrate the edge-side offloading decision.

Adaptive Prediction Sets. Adaptive Prediction Sets (APS) construct a prediction set $\mathcal{S}(x) \subseteq \mathcal{Y}$ instead of producing a single top-1 prediction. Given a user-specified miscoverage level α, APS guarantees

$$\Pr\left(Y_{n+1} \in \mathcal{S}(X_{n+1})\right) \geq 1 - \alpha, \tag{1}$$

where (X_{n+1}, Y_{n+1}) denotes a unseen test point exchangeable with the calibration samples (Romano et al., 2020).

Let $\pi(x)$ be the softmax output of the classifier, and let $\pi_{(k)}(x)$ denote the k-th largest softmax probability. For each calibration sample (x_i, y_i), APS defines the conformity score as the cumulative softmax probability up to the true class:

$$s^{\text{aps}}(x_i, y_i) = \sum_{k=1}^{r(x_i, y_i)} \pi_{(k)}(x_i), \tag{2}$$

where $r(x_i, y_i)$ is the rank of the true label y_i in the descending order of softmax probabilities. Given a calibration set of size n, the APS threshold is computed as

$$\hat{q} = \text{Quantile}\left(\{s^{\text{aps}}(x_i, y_i)\}_{i=1}^{n}; \frac{\lceil (n+1)(1-\alpha) \rceil}{n}\right). \tag{3}$$

At test time, the prediction set is formed by including the smallest number of top-ranked classes whose cumulative probability exceeds $\hat{q}$.

In our framework, APS is used to identify candidate classes when the edge model is uncertain. These candidate classes are then used to construct a class-aware saliency map for patch transmission.

Conformal Risk Control. Conformal Risk Control (CRC) extends the coverage guarantee of conformal prediction to a more general risk-control setting (Angelopoulos et al., 2024). Instead of only controlling whether the true label is included in a prediction set, CRC allows a user-defined bounded loss function to be controlled under a specified risk budget.

Let $\ell(\lambda; x, y) \in [0, B]$ be a loss function parameterized by a threshold λ, where B is a known upper bound satisfying

$$0 \leq \ell(\lambda; x, y) \leq B \tag{4}$$

for all (x, y) and λ. If $\ell(\lambda; x, y)$ is monotonic non-increasing with respect to λ, CRC selects a calibrated threshold $\hat{\lambda}$ using the calibration set such that the expected test-time risk is bounded by a user-specified level α_{crc}:

$$\mathbb{E}\left[\ell(\hat{\lambda}; X_{n+1}, Y_{n+1})\right] \leq \alpha_{\text{crc}}. \tag{5}$$

In this paper, CRC is used to calibrate the edge acceptance threshold. Specifically, the controlled risk corresponds to the event that the edge device accepts its own prediction while the prediction is incorrect. Hence, CRC provides a finite-sample guarantee on the marginal risk of incorrect local acceptance:

$$\mathbb{E}\left[\mathbf{1}\left\{\pi_{(1)}(X_{n+1}) \geq \hat{\lambda} \ \wedge \ \hat{y}_{\text{c}}(X_{n+1}) \neq Y_{n+1}\right\}\right] \leq \alpha_{\text{crc}}. \tag{6}$$

This guarantee should not be interpreted as a guarantee on the overall system accuracy or on the server-side prediction accuracy for offloaded samples.

2.2. Grad-CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) is a saliency visualization method that identifies which spatial regions of an input image contribute to a CNN classifier's prediction (Selvaraju et al., 2017). It uses the gradient of a target class score with respect to the feature activation maps of a convolutional layer to compute a class-discriminative localization map.

Let $A^{(\ell)} \in \mathbb{R}^{C' \times h \times w}$ denote the activation tensor at the last convolutional stage ℓ of a CNN, where C' is the number of channels and $h \times w$ is the spatial resolution. Given the logit z_c of a target class c, Grad-CAM computes the importance weight of channel c' by spatially averaging the gradient:

$$\beta_{c'}^{(c)} = \frac{1}{hw} \sum_{u=1}^{h} \sum_{v=1}^{w} \frac{\partial z_c}{\partial A_{c',u,v}^{(\ell)}}. \tag{7}$$

The class activation map is then obtained by taking a weighted combination of the feature maps:

$$M_c(x) = \text{ReLU}\left(\sum_{c'=1}^{C'} \beta_{c'}^{(c)} A_{c'}^{(\ell)}\right). \tag{8}$$

The ReLU operation preserves regions that positively contribute to the target class. Since the resulting map has a lower spatial resolution than the input image, it is typically upsampled to the input resolution using bilinear interpolation.

We use Grad-CAM as an architecture-agnostic interface between the CNN edge model and the ViT server model by converting CNN-based saliency into ViT patch-level importance scores. Instead of using Grad-CAM only for a single top-1 class, we generalize the Grad-CAM target to an APS-guided candidate class set. This allows the edge CNN to generate a saliency map that reflects multiple plausible classes, which is then converted into a sparse set of ViT patches for communication-efficient server inference.

3. Proposed Method

We propose a statistically calibrated heterogeneous collaborative inference framework for communication-efficient semantic inference. The proposed system consists of a lightweight CNN deployed on the edge device and a high-capacity ViT deployed on the server. Unlike prior attention-aware collaborative inference methods that require VIT-based models in both the edge and the server, our framework enables cross-architecture collaboration by using the CNN not only as an edge-side classifier but also as a saliency generator for ViT patch transmission.

Let f_{c} denote the edge-side CNN classifier and f_{s} denote the server-side ViT classifier. Given an input image x, the edge CNN first produces class logits

$$z = f_{\mathrm{c}}(x), \tag{9}$$

and the corresponding softmax posterior

$$\pi(x) = \mathrm{softmax}(z). \tag{10}$$

Let $\pi_{(1)}(x)$ denote the largest softmax probability and

$$\hat{y}_{\mathrm{c}}(x) = \arg\max_{c} \pi_c(x), \tag{11}$$

denote the edge CNN's top-1 prediction.

To formalize the edge-side reliability decision, let $\lambda \in [0, 1]$ denote the confidence threshold for accepting the edge prediction. For a labeled sample (x, y), we define the edge acceptance loss as

$$\ell(\lambda; x, y) = \mathbf{1}\left\{\pi_{(1)}(x) \geq \lambda \ \wedge \ \hat{y}_{\mathrm{c}}(x) \neq y\right\}. \tag{12}$$

This loss penalizes the failure event where the edge CNN accepts its own prediction but the prediction is incorrect. As λ increases, the edge acceptance criterion becomes stricter, and fewer samples are accepted locally. Therefore, $\ell(\lambda; x, y)$ is monotonic non-increasing in λ, so the CRC framework of Section 2.1 directly applies.

Our goal is to avoid server communication when the edge prediction is sufficiently reliable, while transmitting only semantically important image patches when server inference is required.

3.1. Offline Calibration

Before deployment, we use a held-out calibration set

$$\mathcal{D}_{\mathrm{cal}} = \{(x_i, y_i)\}_{i=1}^{n}, \tag{13}$$

to determine two quantities: the CRC-based edge acceptance threshold $\hat{\lambda}$ and the APS quantile $\hat{q}$. These two calibrated quantities play different roles in the proposed framework. The CRC threshold $\hat{\lambda}$ determines whether the edge prediction can be accepted locally, whereas the APS quantile $\hat{q}$ determines which candidate classes should be considered when constructing the Grad-CAM saliency map for offloaded samples.

For CRC calibration, we use the task-specific loss in Eq. (12), which directly corresponds to the failure event we want to control in the edge-only decision. Using the empirical risk

$$\widehat{R}_n(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \ell(\lambda; x_i, y_i), \tag{14}$$

we select $\hat{\lambda}$ according to the CRC rule described in Section 2.1. The calibrated threshold is then fixed during online inference.

For APS calibration, we compute the APS conformity scores on the same calibration set and obtain the quantile $\hat{q}$. In contrast to conventional conformal classification, we do not use the APS prediction set as the final output. Instead, we use it as a statistically calibrated candidate class set for saliency generation. This distinction is important because the purpose of APS in our framework is not to replace the classifier output, but to prevent the saliency map from depending only on a potentially incorrect top-1 prediction.

3.2. CRC-Calibrated Edge Decision

At test time, the edge CNN first determines whether the input can be processed locally. Given the calibrated threshold $\hat{\lambda}$, the edge accepts its own prediction if

$$\pi_{(1)}(x) \geq \hat{\lambda}. \tag{15}$$

In this case, the final output is

$$\hat{y}(x) = \hat{y}_{\mathrm{c}}(x), \tag{16}$$

and no image data is transmitted to the server.

Under the exchangeability assumption between the calibration data and the test point, the CRC-calibrated threshold $\hat{\lambda}$

controls the marginal risk of incorrect local acceptance, as defined in Eq. (6). This risk is marginal over the test distribution and should be distinguished from the conditional error rate among locally accepted samples. In addition, this guarantee does not cover the server-side prediction for offloaded samples.

If $\pi_{(1)}(x) < \hat{\lambda}$ the edge model regards the input as uncertain and offloads it to the server. However, instead of transmitting the full image, the edge constructs a compact set of semantically informative ViT patches using the procedure described below.

3.3. APS-Guided Class-Aware Grad-CAM

For an offloaded image, the edge first constructs an APS candidate class set $\mathcal{S}(x)$. Let σ_x be the class ordering induced by descending softmax probabilities. Using the calibrated APS quantile $\hat{q}$, we define

$$k^\star(x) = \min\left\{ k : \sum_{j=1}^{k} \pi_{\sigma_x(j)}(x) > \hat{q} \right\}, \tag{17}$$

and obtain the candidate class set

$$\mathcal{S}(x) = \{\sigma_x(1), \sigma_x(2), \ldots, \sigma_x(k^\star(x))\}. \tag{18}$$

The key idea is to compute Grad-CAM not from a single top-1 class, but from the APS candidate class set. For each class $c \in \mathcal{S}(x)$, we assign a normalized weight

$$w_c(x) = \frac{\pi_c(x)}{\sum_{c' \in \mathcal{S}(x)} \pi_{c'}(x)}. \tag{19}$$

Then, instead of using a single class logit as the Grad-CAM target, we define an aggregated target logit

$$g(x) = \sum_{c \in \mathcal{S}(x)} w_c(x) z_c. \tag{20}$$

Grad-CAM is then applied to $g(x)$ with respect to the last convolutional feature map of the edge CNN.

The intuition behind this aggregated target is to consolidate the spatial regions that each plausible candidate class deems important, weighted by the model's confidence in that class.

Let $A^{(\ell)} \in \mathbb{R}^{C' \times h \times w}$ denote the activation tensor at the last convolutional stage. The channel importance weight is computed as

$$\beta_{c'} = \frac{1}{hw} \sum_{u=1}^{h} \sum_{v=1}^{w} \frac{\partial g(x)}{\partial A^{(\ell)}_{c',u,v}}, \tag{21}$$

and the class-aware saliency map is obtained by

$$M(x) = \mathrm{Up}\left(\mathrm{ReLU}\left(\sum_{c'=1}^{C'} \beta_{c'} A^{(\ell)}_{c'} \right) \right). \tag{22}$$

Here, $\mathrm{Up}(\cdot)$ denotes bilinear upsampling to the input image resolution. The resulting map is then min–max normalized to $[0, 1]$.

This design allows the saliency map to reflect multiple plausible classes rather than only the top-1 prediction. Therefore, when the edge CNN is uncertain, the transmitted patches are less likely to be biased toward an incorrect class-specific region.

3.4. Saliency-to-Patch Conversion

The server ViT processes the input as a sequence of image patches. Let the ViT patch grid size be $P \times P$, where $P = H/d$ for image resolution $H \times H$ and patch size d. To align the CNN saliency map with the ViT patch grid, we apply adaptive average pooling

$$\widetilde{M}(x) = \mathrm{Norm}\left(\mathrm{AvgPool}_{P \times P}(M(x))\right), \tag{23}$$

where $\widetilde{M}(x) \in [0, 1]^{P \times P}$ represents patch-level semantic importance.

Given a patch selection threshold τ, the edge selects the patch index set

$$\mathcal{P}(x) = \left\{ p \in \{1, \ldots, P^2\} : \widetilde{M}_p(x) \geq \tau \right\}. \tag{24}$$

If no patch satisfies the threshold, we select the most salient patch to ensure that the server receives a non-empty input

$$\mathcal{P}(x) = \left\{ \arg\max_p \widetilde{M}_p(x) \right\}. \tag{25}$$

The edge transmits only the selected image patches $\{x_p\}_{p \in \mathcal{P}(x)}$ and their corresponding patch indices $\mathcal{P}(x)$ to the server. Since the patch index overhead is small compared to the image patch payload, the communication cost is mainly determined by the number of transmitted patches.

3.5. Server-Side Inference

After receiving the selected patches and their indices, the server ViT performs final classification using only the transmitted patch tokens. The server prediction is given by

$$\hat{y}_{\mathrm{s}}(x) = \arg\max_c f_{\mathrm{s},c}(x; \mathcal{P}(x)). \tag{26}$$

The final system output is therefore

$$\hat{y}(x) = \begin{cases} \hat{y}_{\mathrm{c}}(x), & \pi_{(1)}(x) \geq \hat{\lambda}, \\ \hat{y}_{\mathrm{s}}(x), & \pi_{(1)}(x) < \hat{\lambda}. \end{cases} \tag{27}$$

Thus, the proposed inference rule combines CRC-calibrated local acceptance with APS-guided saliency-based patch offloading in a heterogeneous CNN-to-ViT architecture.

4. Experiments

4.1. Experimental Setup

We evaluate the proposed framework on the ImageNet-1K validation set (Deng et al., 2009). The validation set is divided into 10,000 calibration images and 40,000 test images. All images are center-cropped to 224×224, which corresponds to $14 \times 14 = 196$ image patches. The proposed framework employs a pretrained EfficientNet-B0 as the edge model and DeiT-Base as the server model (Tan & Le, 2019; Touvron et al., 2021). The CRC risk budget is fixed at $\alpha_{\text{crc}} = 0.05$, and the APS miscoverage level is fixed to $\alpha = 0.05$. Grad-CAM is applied to the last convolutional layer of EfficientNet-B0.

We compare the proposed framework with three patch-selection baselines: random patch selection, top-k patch selection, and attention-sum thresholding. For all baselines, DeiT-Tiny is used as the edge model, and the entropy threshold is fixed at $\eta = 0.8$, following the evaluation setting of prior attention-aware collaborative inference work (Im et al., 2024). The server-only baseline represents the upper-bound setting in which all 196 patches are transmitted to DeiT-Base. Communication cost is measured as the ratio of the total transmitted bytes to the total raw image bytes over the test set, following the same definition.

4.2. Experimental Results

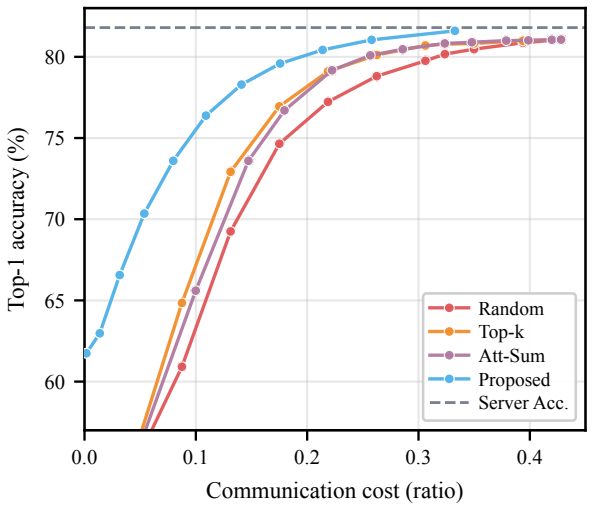

Figure 1. Communication-accuracy trade-off on ImageNet dataset.

Accuracy. Figure 1 shows the communication–accuracy trade-off of the proposed framework and the three patch-selection baselines. The trade-off curves are generated by varying the patch-selection threshold τ in the proposed method, the attention-sum threshold δ_{sum} in Att-Sum, and the number of transmitted patches k in Top-k and Random.

Table 1. Accuracy–communication trade-off under two evaluation settings. Communication cost is normalized by full-image transmission to the DeiT-Base server, which achieves 81.80% Top-1 accuracy.

Method	Best Acc.		Within 1 pp Gap	
	Cost	Acc. (%)	Cost	Acc. (%)
Random	0.43	81.07	0.39	80.86
Top-K	0.43	81.07	0.35	80.82
Attention Sum	0.43	81.06	0.32	80.84
Proposed	**0.33**	**81.60**	**0.26**	**81.05**

Table 1 summarizes two representative operating points selected from Figure 1. At the best-accuracy operating point, the proposed method achieves 81.60% Top-1 accuracy with a communication cost of 0.33. This corresponds to a 67% reduction in communication cost relative to full-image transmission, while incurring only a 0.20 pp accuracy loss compared with the server-only DeiT-Base model, which achieves 81.80% Top-1 accuracy. In contrast, the baseline methods require a communication cost of 0.43 to achieve around 81.07% Top-1 accuracy.

When a 1 pp accuracy gap from the server-only model is allowed, the proposed method achieves 81.05% Top-1 accuracy with a communication cost of 0.26, corresponding to a 74% reduction relative to full-image transmission. In both evaluation settings, the proposed method achieves the lowest communication cost while maintaining the highest Top-1 accuracy among the compared methods. These results demonstrate the effectiveness of combining CRC-calibrated local acceptance with APS-guided saliency-based patch transmission.

Transmission Latency Analysis. We further evaluate the practical impact of communication cost reduction in terms of transmission latency under realistic wireless conditions. Following (Im et al., 2024), we estimate the data size of a full image as 147 KB based on a $224 \times 224 \times 3$ byte representation. We consider upload data rates of 1, 8, and 20 Mb/s, consistent with the setting in (Zhang et al., 2023).

Table 2. Settings (a) and (b) correspond to the best-accuracy and within 1 pp gap cases.

Setting	Method	Acc. (%)	Upload Data Rate		
			1 Mb/s	8 Mb/s	20 Mb/s
(a)	Server-only	81.80	1176.00	147.00	58.80
	Att-Sum	81.06	503.33	62.92	25.17
	Proposed	81.60	**391.26**	**48.91**	**19.56**
(b)	Server-only	81.80	1176.00	147.00	58.80
	Att-Sum	80.83	380.55	47.57	19.03
	Proposed	81.05	**303.29**	**37.91**	**15.16**

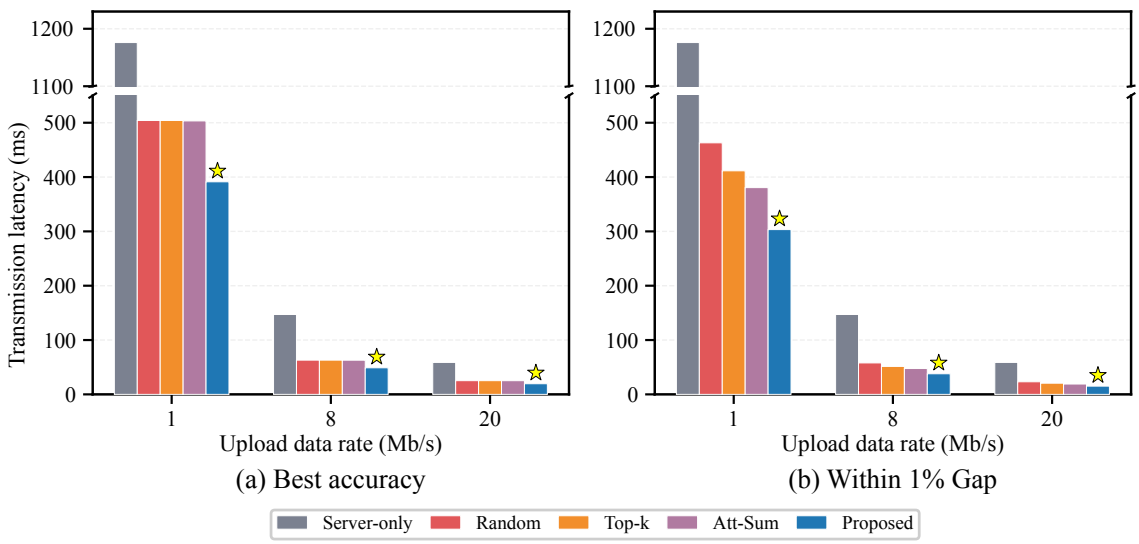

Figure 2. Transmission latency comparison across upload data rates on the ImageNet dataset. (a) Each method operates at its best-accuracy point. (b) Each method operates at the point where its top-1 accuracy is within 1 pp of the server-only baseline ($\geq$ 80.8%). Stars denote the operating points reported in Figure 1.

Figure 2 and Table 2 show the transmission latency at the representative operating points selected from Figure 1. Across all upload data rates, the proposed method achieves the lowest latency in both the best-accuracy setting and the setting within 1 pp of the server-only baseline.

5. Conclusion

In this paper, we proposed a risk-calibrated semantic transmission framework for communication-efficient heterogeneous collaborative inference. The proposed method combines a lightweight CNN on the edge device with a high-capacity ViT on the server, enabling CNN-to-ViT collaborative inference. Conformal Risk Control (CRC) is used to calibrate the risk of incorrect local acceptance, while Adaptive Prediction Sets (APS) and Grad-CAM are used to selectively transmit only semantically informative image patches when server inference is required. Experimental results on ImageNet show that the proposed method maintains high Top-1 accuracy with lower communication cost than existing patch-selection baselines. In particular, under the setting within a 1 pp accuracy gap from the server-only baseline, the proposed method reduces communication cost by 74% while achieving 81.05% Top-1 accuracy. The transmission latency analysis also shows that the proposed method achieves the lowest latency across different uplink data rates, demonstrating its effectiveness in improving both reliability and communication efficiency.

References

Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L., and Schuster, T. Conformal risk control. In *International Conference on Learning Representations*, 2024.

Chen, J. and Ran, X. Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8):1655–1674, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Im, J., Kwon, N., Park, T., Woo, J., Lee, J., and Kim, Y. Attention-aware semantic communications for collaborative inference. *IEEE Internet of Things Journal*, 11(22): 37008–37020, 2024. doi: 10.1109/JIOT.2024.3440313.

Jung, B. C. Toward artificial intelligence-native 6G services. *IEEE Vehicular Technology Magazine*, 2024.

Kang, Y., Hauswald, J., Gao, C., Rovinski, A., Mudge, T. N., Mars, J., and Tang, L. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge.

In *Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 615–629, 2017. doi: 10.1145/3037697.3037698.

Li, E., Zhou, Z., and Chen, X. Edge intelligence: On-demand deep learning model co-inference with device-edge synergy. In *Proceedings of the 2018 Workshop on Mobile Edge Communications*, pp. 31–36, 2018. doi: 10.1145/3229556.3229562.

Ren, W.-Q., Qu, Y.-B., Dong, C., Jing, Y.-Q., Sun, H., Wu, Q.-H., and Guo, S. A survey on collaborative dnn inference for edge intelligence. *Machine Intelligence Research*, 20(3):370–395, 2023. doi: 10.1007/s11633-022-1391-7.

Romano, Y., Sesia, M., and Candès, E. J. Classification with valid and adaptive coverage. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3581–3591, 2020.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017. doi: 10.1109/ICCV.2017.74.

Tan, M. and Le, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114. PMLR, 2019.

Teerapittayanon, S., McDanel, B., and Kung, H. T. Distributed deep neural networks over the cloud, the edge and end devices. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 328–339, 2017. doi: 10.1109/ICDCS.2017.226.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jegou, H. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 2021.

Wu, J., Li, R., An, X., Peng, C., Liu, Z., Crowcroft, J., and Zhang, H. Toward native artificial intelligence in 6G networks: System design, architectures, and paradigms. *arXiv preprint arXiv:2103.02823*, 2021.

Zhang, X., Mounesan, M., and Debroy, S. EFFECT-DNN: Energy-efficient edge framework for real-time DNN inference. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 10–20, 2023. doi: 10.1109/WoWMoM57956.2023.00015.

Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., and Zhang, J. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8):1738–1762, 2019.

A. Detailed Algorithms

Algorithm 1 describes the offline calibration procedure. Algorithm 2 describes the online inference procedure.

Algorithm 1 Offline calibration

Require: Edge CNN f_c, calibration set $\mathcal{D}_{\mathrm{cal}} = \{(x_i, y_i)\}_{i=1}^{n}$, CRC budget α_{crc}, APS miscoverage α
Ensure: CRC threshold $\hat{\lambda}$, APS quantile $\hat{q}$
$\mathcal{C} \leftarrow \emptyset, \quad \mathcal{Q} \leftarrow \emptyset$
for $i = 1, \ldots, n$ **do**
$\pi_i \leftarrow \mathrm{softmax}(f_c(x_i))$
$s_i \leftarrow \pi_{i,(1)}$
$\ell_i \leftarrow \mathbf{1}\{\arg\max_c \pi_{i,c} \neq y_i\}$
$\mathcal{C} \leftarrow \mathcal{C} \cup \{(s_i, \ell_i)\}$
$r_i \leftarrow \mathrm{rank}(y_i)$ in descending order of π_i
$s_i^{\mathrm{aps}} \leftarrow \sum_{k=1}^{r_i} \pi_{i,(k)}$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{s_i^{\mathrm{aps}}\}$
end for
$\hat{\lambda} \leftarrow$ smallest λ such that $\widehat{R}_n(\lambda) \leq \alpha_{\mathrm{crc}} - \frac{B - \alpha_{\mathrm{crc}}}{n}$ on $\mathcal{C}$
$\hat{q} \leftarrow \mathrm{Quantile}\left(\mathcal{Q}; \frac{\lceil (n+1)(1-\alpha) \rceil}{n}\right)$
return $(\hat{\lambda}, \hat{q})$

Algorithm 2 Online inference with class-aware patch transmission

Require: Test image x, edge CNN f_{c}, server ViT f_{s}, thresholds $(\hat{\lambda}, \hat{q}, \tau)$, patch grid size P
Ensure: Final prediction $\hat{y}$
$z \leftarrow f_c(x)$
$\pi \leftarrow \mathrm{softmax}(z)$
if $\pi_{(1)} \geq \hat{\lambda}$ **then**
$\hat{y} \leftarrow \arg\max_c \pi_c$
return $\hat{y}$
end if
$k^\star \leftarrow \min\left\{k : \sum_{j=1}^{k} \pi_{(j)} > \hat{q}\right\}$
$\mathcal{S}(x) \leftarrow \{\sigma_x(1), \ldots, \sigma_x(k^\star)\}$
$w_c \leftarrow \pi_c / \sum_{c' \in \mathcal{S}(x)} \pi_{c'}$ for all $c \in \mathcal{S}(x)$
$g \leftarrow \sum_{c \in \mathcal{S}(x)} w_c z_c$
Backpropagate g to obtain $\partial g / \partial A^{(\ell)}$
Compute Grad-CAM map $M(x)$ and apply min–max normalization
$\widetilde{M}(x) \leftarrow \mathrm{Norm}\left(\mathrm{AvgPool}_{P \times P}(M(x))\right)$
$\mathcal{P}(x) \leftarrow \{p : \widetilde{M}_p(x) \geq \tau\}$
if $\mathcal{P}(x) = \emptyset$ **then**
$\mathcal{P}(x) \leftarrow \{\arg\max_p \widetilde{M}_p(x)\}$
end if
Transmit $\{x_p\}_{p \in \mathcal{P}(x)}$ and $\mathcal{P}(x)$ to the server
$\hat{y} \leftarrow \arg\max f_s(x; \mathcal{P}(x))$
return $\hat{y}$