基于轻量化Swin Transformer的资源自适应语义压缩策略

郑飞[1]，吴东颖[1]，李世超[1]，仇洪冰[1]，邵苏杰[2]，喻鹏[2]，丰雷[2]，赵继龙[1]
（1.桂林电子科技大学认知无线电与信息处理省部共建教育部重点实验室，广西 桂林 541004；
2.北京邮电大学网络与交换技术全国重点实验室，北京 100876）

摘要：在面向多终端的下行语义通信场景中，由于终端算力的差异性，采用统一的语义压缩比提取语义信息，会造成低算力终端难以完成语义解码、高算力终端的算力资源未充分利用和语义数据不够精练的问题。为解决上述问题，针对有限的终端算力资源和链路带宽，提出了一种基于轻量化Swin Transformer的资源自适应语义压缩策略，采用门控网络和稀疏注意力机制改进了语义编码器结构，为终端定制差异化语义压缩比，以最小化系统总能耗为优化目标，构建联合算力和带宽分配模型，并采用近端策略优化（PPO）算法求解优化问题。仿真结果表明，与固定语义压缩比方案相比，所提策略的系统总能耗降低了39%。

关键词：语义通信；Swin Transformer；稀疏注意力机制；近端策略优化算法
中图分类号：TN919.8
文献标志码：A
doi：10.11959/j.issn.1000–436x.TXXB260128

Lightweight Swin Transformer-based resource-adaptive semantic compression strategy

Zheng Fei[1], Wu Dongying[1], Li Shichao[1], Qiu Hongbing[1], Shao Sujie[2], Yu Peng[2], Feng Lei[2], Zhao Jilong[1]
1. Ministry of Education Key Lab. of Cognitive Radio and Information Processing, Guilin University of Electronic Technology, Guilin 541004, China
2. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract: In multi-terminal downlink semantic communication scenarios, employing a uniform semantic compression ratio caused difficulty in semantic decoding for low-computing-power terminals, as well as underutilized computing power resources and insufficiently refined semantic data for high-computing-power terminals. To address this issue under constraints of limited terminal computing power and link bandwidth, a lightweight Swin Transformer-based resource-adaptive semantic compression strategy was proposed. The semantic encoder integrated a gating network and a sparse attention mechanism to customize differentiated semantic compression ratios for individual terminals. A joint computing and bandwidth allocation model was formulated to minimize total system energy consumption. The proximal policy optimization (PPO) algorithm was employed to solve the optimization problem. Simulation results demonstrate that, compared with fixed compression ratio schemes, the proposed strategy reduces total system energy consumption by 39%.

Key words: semantic communication, Swin Transformer, sparse attention mechanism, proximal policy optimization algorithm

收稿日期：2026–03–18；**修回日期：**2026–04–26
通信作者：李世超，shichaoli@guet.edu.cn
基金项目：国家自然科学基金资助项目（No.62361016）；广西重点研发计划基金资助项目（No.GuikeAB25069438）
Foundation Items: The National Natural Science Foundation of China (No.62361016), The Guangxi Key Research and Development Program (No.GuikeAB25069438)

0　引言

随着自动驾驶、虚拟现实等新兴应用的快速发展，大量终端并发接入无线通信网络，图像和视频等视觉数据的传输需求激增，在有限带宽和能源约束下为高密度无线接入终端传输海量视觉数据是一个新挑战。在传统通信范式中，数据的发送和接收主要依赖于符号级别的准确传输，忽略了数据的含义，导致大量冗余数据在网络中传输，浪费了带宽和能量。因此，需要革命性的通信范式来满足新兴的业务需求[1]。语义通信通过提取和传输数据的含义，而非完整的原始数据，能够显著减少传输数据量，提高通信效率，成为一种新兴的技术方向。然而，在实际部署场景中，终端算力呈现显著差异性：一方面，自动驾驶汽车等终端具备较高算力，可支持复杂的语义编解码；另一方面，大量物联网终端、边缘传感器等算力较为有限，难以承担计算密集的语义处理任务。这种差异性使语义通信系统在模型设计与资源调度方面面临新的挑战。

早期的语义通信系统利用卷积神经网络构建语义编码器和解码器，如 Bourtsoulatze 等[2]提出的联合信源信道编码方案。然而，卷积神经网络对长距离依赖关系的捕获能力有限，难以获取全局语义信息。为了克服该限制，后续研究将具有自注意力机制的 Transformer 架构引入图像语义通信中。其中，Swin Transformer 的层次化结构能够捕获从局部到全局的多尺度语义信息，滑动窗口自注意力机制显著降低了计算复杂度，移位窗口设计保证了跨窗口信息交互，是图像语义通信的理想选择。文献[3]设计了一种 SwinJSCC 架构用于图像语义传输，以 Swin Transformer 为骨干网络，与基于卷积神经网络的联合信源信道编码方法相比，实现了显著的性能增益。文献[4]通过在 Swin Transformer 模块间引入残差连接，进一步增强了图像语义特征提取能力。然而，语义编码依赖神经网络模型，需要消耗大量计算资源[5]。随着语义通信系统的发展，语义编码模型复杂度不断提升，系统能耗问题日益凸显，成为制约语义通信系统大规模部署的关键挑战。现有研究从不同层面对语义通信系统进行节能优化。一类研究尝试从模型层面对 Transformer 结构进行轻量化设计，包括 Token 剪枝（如动态 Token 剪枝（dynamic token pruning, DToP[6]））、改进注意力机制（如 EdgeViTs[7]、SparseFormer[8]、原生稀疏注意力[9]），这些方法均在保持语义质量的前提下降低了计算开销。另一类研究设计资源分配机制以优化资源利用率，如文献[10]利用知识图谱提取语义信息，采用统一的语义压缩比，迭代优化压缩比和功率分配以最小化网络总能耗；文献[11]利用近端策略优化（proximal policy optimization，PPO）算法联合优化系统压缩比、信道分配和功率分配，以提升系统频谱资源利用率并保证图像传输质量；文献[12]通过联合优化语义提取因子、通信与计算资源，在满足时延和任务处理速率的约束下有效降低了系统能耗。尽管上述研究在模型轻量化和资源优化方面取得了重要进展，但其模型计算复杂度和语义压缩比在服务过程中保持固定，在多终端算力差异的场景中会导致低算力终端解码困难、高算力终端资源未充分利用和语义数据不够精练的问题。

针对上述问题，部分研究开始探索自适应机制，通过动态调整资源分配策略或语义编码器结构，使系统能够根据终端算力差异提供定制化服务。一类研究通过将语义模型的计算任务在终端与边缘服务器间灵活分配以缓解低算力终端的计算压力，例如，文献[13]通过细粒度拆分 Vision transformer，并在终端与边缘设备间动态卸载推理任务以降低推理延迟；文献[14]将用户任务的语义信息卸载到边缘服务器计算，并采用多智能体 PPO 算法协调资源管理；文献[15]利用 D2D 通信辅助终端间协同计算，结合有向图卷积网络与多智能体强化学习（reinforcement learning，RL）设计协作拓扑与资源分配算法，有效降低了端侧协同的通信开销。另一类研究设计具备可调节能力的语义编码器，从而为不同终端提供差异化的语义表示，例如，文献[16]提出了一种基于目标嵌入的语义编码器，设计目标嵌入模块引导编码器提取特定解码器的语义特征，并利用信噪比信息和网络通信状态编码语义特征向量并调整向量长度；文献[17]根据接收端恢复图像能力差异，提出一种基于图像特征贡献因子的非均匀量化方法，动态分配量化比特，在保证图像重建质量的同时降低了带宽需求；文献[18]通过动态知识蒸馏为异构边缘终端定制语义提取模型，从而在保证语义准确性的同时降低模型复杂度和网络通信开销；文献[19]设计了一个面向多基站多终端的系统，通过引入一种推理早退机制来动态

调整神经网络计算深度，降低了计算开销。还有一类研究通过联合优化语义压缩与资源分配，在一定程度上提高了系统资源效率，例如，文献[20]将压缩比选择和通信资源分配问题拆分为两个子问题并迭代求解以保证多用户的服务质量；文献[21]定量分析了压缩比与计算比之间的关系，并通过联合优化压缩比、计算资源分配、卸载时间以及深度神经网络层选择，在无线资源受限条件下降低终端能耗并提高系统资源利用率。虽然上述研究从不同角度探索了终端算力差异的适配方法，但前两类研究仍主要针对单一维度进行自适应设计，难以同时兼顾语义模型结构与系统资源配置的协同优化需求。尽管第三类研究将语义压缩与资源分配纳入统一优化框架，但其在求解过程中仍采用交替迭代与变量分解等分步优化策略，未能充分考虑计算资源与通信资源之间的相互制约关系，从而限制了系统整体能效的进一步提升。

受上述研究启发，本文将语义压缩比调节、算力与带宽分配纳入统一的端到端优化框架中，协同设计了可调节的语义编码器结构与资源调度策略，从而在适配终端算力差异的同时提升系统能效。RL 中的 PPO 算法通过限制策略更新幅度保证了训练稳定性，并在混合动作空间中具有良好适用性，是求解该优化问题的有效手段。然而，在多终端场景下，状态空间维度随终端数量线性增长、无关信息冗余严重，导致 PPO 算法面临决策效率下降、收敛缓慢的问题，亟须一种轻量化的求解方法以加速策略学习过程。

综上所述，本文提出了一种基于轻量化 Swin Transformer 的资源自适应语义压缩策略，同时用稀疏注意力改进了 PPO 算法。该策略利用 Swin Transformer 构建语义编码器，引入门控网络实现语义压缩比的动态调整，改进 Swin Transformer 的注意力机制以降低计算复杂度，并在系统层面构建联合算力与带宽优化模型，实现最小化系统总能耗。本文主要贡献总结如下。

（1）提出了一种基于轻量化 Swin Transformer 的资源自适应语义压缩策略，设计了门控网络以动态调整语义编码器深度，为终端定制差异化语义压缩比，同时采用了“全局压缩+Top-k 选择”的稀疏注意力机制改进了 Swin Transformer，降低了模型计算复杂度。

（2）构建了“基站（base station, BS）编码–下行传输–终端解码”全过程能耗模型，并在此基础上建立以系统总能耗最小化为目标的联合优化问题，将联合语义压缩比、算力与带宽分配统一纳入系统能耗优化框架中。

（3）将上述优化问题转化为马尔可夫决策过程（Markov decision process, MDP），并采用 PPO 算法求解。同时，设计了一种稀疏注意力改进的 PPO 算法，在其策略（Actor）网络中引入稀疏注意力模块，以提高 PPO 算法的策略学习效率与收敛速度。

1 系统模型

考虑一个单 BS、多终端下行传输的语义通信网络架构，包括一个 BS 和 U 个终端，终端集合用 $\mathcal{U}=\{1,2,\cdots,U\}$ 表示，如图 1 所示。BS 使用语义编码器提取原始图像数据的语义信息并传输给终端。终端接收到语义数据后，利用语义解码器从语义数据中重建图像。由于算力的差异性，终端的语义解码能力不同，为保证终端高效完成语义解码，BS 需要在语义编码时为终端定制差异化语义压缩比。定义终端 u 的语义压缩比 ρ_u 为语义编码器输入的原始图像数据大小 d_u 和输出的语义数据大小 s_u 的比值[22]，即 $\rho_u=\dfrac{d_u}{s_u}$。

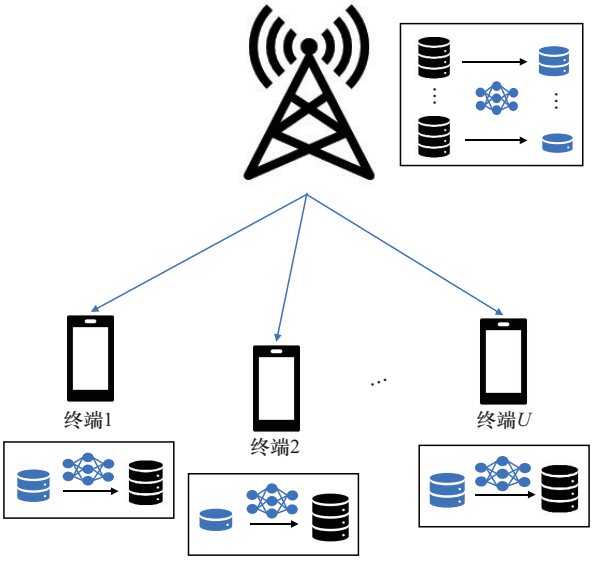


图 1 下行语义通信网络架构

1.1 语义编码器结构

采用 Swin Transformer 模型作为本文的语义通

信系统模型，并且为了适应终端算力差异性，用门控网络和稀疏注意力机制改进了一种语义压缩比可调的语义编码器结构。

1.1.1　门控网络设计

在 Swin Transformer 的阶段 3 设计了一种门控网络，根据终端算力调整参与计算的 Swin Transformer block 个数，即调整语义编码器深度，从而实现自适应的语义压缩，如图 2 所示。

更深的编码器会消耗更多的算力，但可以从原始图像中提取更精炼的语义数据，即语义压缩比越高。阶段 3 的 Swin Transformer 块集合为 $\mathcal{L}=\{1,2,\cdots,L\}$，参与计算的 Swin Transformer 块个数 l 与语义压缩比 ρ 呈正相关。

$$\rho = f(l) \tag{1}$$

其中，$f(\cdot)$ 是 ρ 关于 l 的非线性正相关函数。BS 语义编码的计算量 ε_{enc} 与 l 的关系为：

$$\varepsilon_{\text{enc}} = \alpha l + C \tag{2}$$

其中，α 是每个块的计算参数，C 是固定计算开销。由于 l 与 ρ 呈正相关，因此 ε_{enc} 与 ρ 也呈正相关，即 $\rho \propto \varepsilon_{\text{enc}}$，记作 $\varepsilon_{\text{enc}}(\rho)$。

更少的语义数据会占用更少的带宽，但语义解码会消耗更多算力。相应地，部署在终端的语义解码器采用与语义编码器对称的结构，且语义解码的计算量与语义编码相同，即 $\varepsilon_{\text{dec}}(\rho)=\varepsilon_{\text{enc}}(\rho)$。

1.1.2　稀疏注意力机制设计

为节省计算开销，窗口内采用“全局压缩+Top-k 选择”的稀疏注意力机制，包括两部分：全局压缩注意力机制和 Top-k 选择注意力机制，如图 3 所示。

（1）全局压缩注意力机制

首先，将长度为 N 的 token 序列按顺序均分成 M 个块，每个块包含 $m=\dfrac{N}{M}$ 个 token（$M<N$），对块内所有键（Key）和值（Value）向量分别取均值得到每个压缩块的 Key 和 Value。第 j 个压缩块的压缩 Key K_j^C 和压缩 Value V_j^C 可分别表示为：

$$K_j^C = \frac{1}{m}\sum_{i=(j-1)m+1}^{jm} K_i \tag{3}$$

$$V_j^C = \frac{1}{m}\sum_{i=(j-1)m+1}^{jm} V_i \tag{4}$$

经此步骤，由原来 N 个键值对减少为 M 个压缩块的键值对。这些压缩块能够捕捉片段内的高层语义和总体概要。接下来，对于每个查询（Query）（对应输入序列的第 i 个 token），计算它与所有压缩 Key、压缩 Value 的注意力分数：

$$\text{ComAtten}(Q_i) = \sum_{j=1}^{M} \alpha_{ij}^C V_j^C \tag{5}$$

其中，$\alpha_{ij}^C = \text{Softmax}\left(Q_i \cdot \dfrac{K_j^C}{\sqrt{d}}\right)$ 是 Q_i 对第 i 个压缩块的注意力权重，d 为隐藏维度。

（2）Top-k 选择注意力机制

在全局压缩过程中，如果某些重要的细节在平

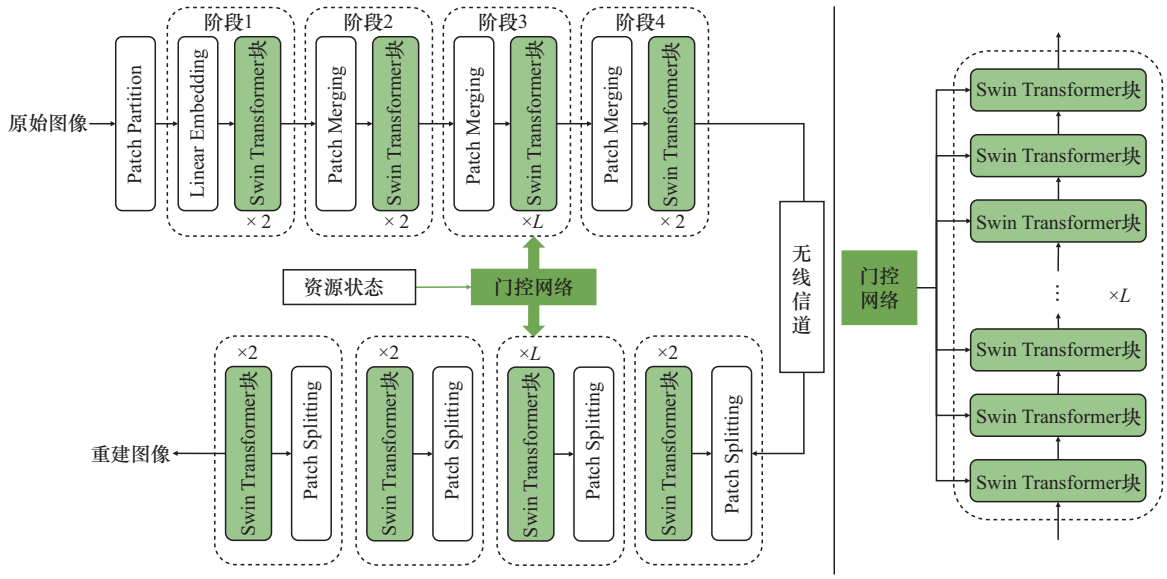
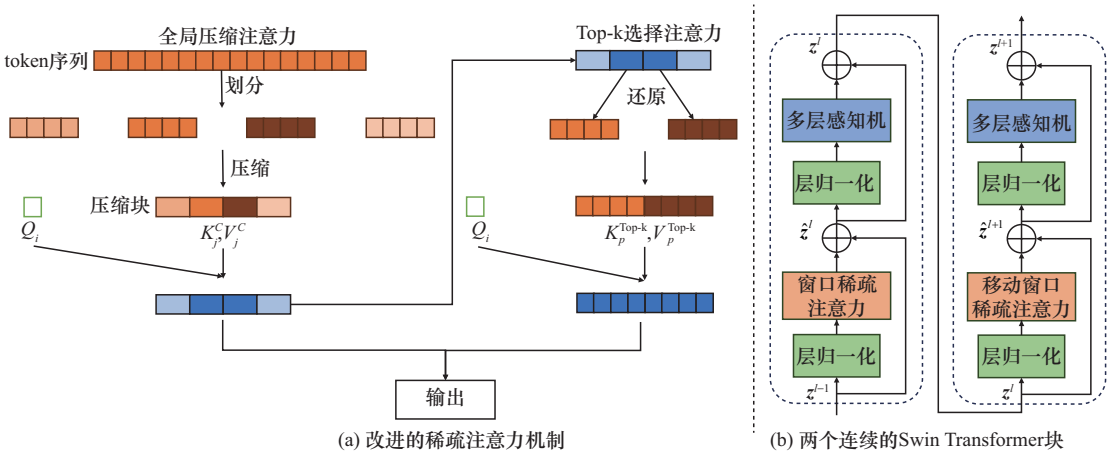


图 2　门控网络改进的语义编解码器结构



(a) 改进的稀疏注意力机制　　(b) 两个连续的Swin Transformer块

图3　改进的稀疏注意力机制

均时被稀释，那么只靠全局压缩注意力表示不足以重建原始序列的信息。因此，结合Top-k选择注意力机制，可以进一步保留关键细节。

选出几个“最重要”的压缩块并恢复其中原始的token参与注意力计算。具体而言，根据全局压缩注意力机制求得的注意力分数对压缩块进行排序，选出分数最高的前k个压缩块（$k<M$），它们被认为包含了关键信息。假设选中的k个压缩块的集合为$\mathcal{K}=\{1,2,\cdots,k\}$，$\mathcal{K}$中恢复的原始token的集合为$\mathcal{P}=\{p_{11},\cdots,p_{1r},\cdots,p_{k1},\cdots,p_{kr}\}$，Key和Value的集合分别为$K^{\text{Top-k}}=\{K_p^{\text{Top-k}}\mid p\in\mathcal{P}\}$，$V^{\text{Top-k}}=\{V_p^{\text{Top-k}}\mid p\in\mathcal{P}\}$。每个$Q_i$与集合$\mathcal{P}$上计算注意力：

$$\text{TopAtten}(Q_i)=\sum_{p\in\mathcal{P}}\alpha_{ip}^{\text{Top-k}}V_p^{\text{Top-k}} \tag{6}$$

其中，$\alpha_{ip}^{\text{Top-k}}=\text{Softmax}\left(Q_i\cdot\dfrac{K_p^{\text{Top-k}}}{\sqrt{d}}\right)$是$Q_i$对$\mathcal{P}$中的token p的注意力权重。

本文改进的“全局压缩+Top-k选择”的稀疏注意力机制只针对全局压缩后的M个压缩块和k个压缩块内的token计算，减少了语义编解码的算力开销。

1.2　时延与能耗

BS语义编码终端u的图像数据时的计算时延t_u^{enc}可表示为：

$$t_u^{\text{enc}}=\frac{\varepsilon_{\text{enc}}(\rho_u)d_u}{\varepsilon_B f_u^B} \tag{7}$$

其中，d_u表示终端u的图像数据大小，$\varepsilon_{\text{enc}}(\rho_u)$表示编码器处理1 bit所需的浮点运算次数（与语义压缩比ρ_u正相关），ε_B表示BS每个GPU周期可执行的浮点运算次数，f_u^B表示BS语义编码终端u的数据时使用的GPU周期数。BS语义编码终端u的图像数据时的计算能耗表示为：

$$E_u^{\text{enc}}=\kappa_B t_u^{\text{enc}}\left(f_u^B\right)^3=\kappa_B\frac{\varepsilon_{\text{enc}}(\rho_u)d_u}{\varepsilon_B}\left(f_u^B\right)^2 \tag{8}$$

其中，κ_B是有效开关电容系数，具体取决于BS的GPU芯片架构。

终端u语义解码时的计算时延t_u^{dec}表示为：

$$t_u^{\text{dec}}=\frac{\varepsilon_{\text{dec}}(\rho_u)s_u}{\varepsilon_u f_u} \tag{9}$$

其中，s_u表示终端u接收到的语义数据大小，$\varepsilon_{\text{dec}}(\rho_u)$表示解码器处理1 bit所需的浮点运算次数（与语义压缩比ρ_u呈正相关），ε_u表示终端u每个GPU周期可执行的浮点运算次数，f_u表示终端u语义解码使用的GPU周期数。终端u语义解码的能耗表示为：

$$E_u^{\text{dec}}=\kappa_U\frac{\varepsilon_{\text{dec}}(\rho_u)s_u}{\varepsilon_u}\left(f_u\right)^2 \tag{10}$$

其中，κ_U是有效开关电容系数，具体取决于用户的GPU芯片架构。

BS与终端之间的总链路带宽为B，假设用户u的语义数据s_u占用的链路带宽为B_u，那么BS到终端u的传输速率为：

$$R_u=B_u\text{lb}\left(1+\frac{p_{\text{BS}}h_u}{n_0B_u}\right) \tag{11}$$

其中，p_{BS}是BS的发射功率，h_u是BS与终端u之间的信道增益，n_0是噪声的功率谱密度。BS给终

端 u 传输的语义数据大小为 $s_u = \dfrac{d_u}{\rho_u}$，传输的语义符号通过二进制量化映射到比特上，因此物理层传输仍然遵循香农经典信息论。因此 BS 向终端 u 传输语义数据的时延可以表示为：

$$t_u^{\text{trans}} = \frac{s_u}{R_u} \tag{12}$$

BS 向终端 u 传输语义数据的能耗可以表示为：

$$E_u^{\text{trans}} = \frac{s_u}{R_u} p_{\text{BS}} \tag{13}$$

因此系统的总能耗可以表示为：

$$E_{\text{total}} = \sum_{u=1}^{U} (E_u^{\text{enc}} + E_u^{\text{trans}} + E_u^{\text{dec}}) \tag{14}$$

1.3　问题描述

以最小化系统总能耗为优化目标，同时考虑 BS 的算力消耗、链路带宽占用、任务的最大可容忍时延、语义压缩比和图像重建质量（PSNR）[23] 等约束，问题可表述为：

$$\min_{\rho_u, f_u, B_u} E_{\text{total}} \tag{15}$$

$$\begin{aligned}
\text{s.t.}\quad & \text{C1: } \sum_{u=1}^{U} f_u^B \leqslant f_{\max}^B \\
& \text{C2: } 0 \leqslant f_u \leqslant f_u^{\max} \\
& \text{C3: } \sum_{u=1}^{U} B_u \leqslant B \\
& \text{C4: } 0 < t_u^{\text{enc}} + t_u^{\text{trans}} + t_u^{\text{dec}} \leqslant \tau_{\max} \\
& \text{C5: } \rho_u \geqslant 1 \\
& \text{C6: } \text{PSNR}_u \geqslant \text{PSNR}_{\min}
\end{aligned} \tag{16}$$

其中，约束 C1 表示 BS 语义编码终端数据消耗的算力之和不超过 BS 总算力 $f_{\max}^B$；约束 C2 表示终端 u 语义解码消耗的算力不超过终端 u 的最大算力 $f_u^{\max}$；约束 C3 表示终端的语义数据占用带宽之和不超过链路总带宽 B；约束 C4 表示终端 u 的图像语义编解码和传输数据的总时延不超过任务的最大可容忍时延 $\tau_{\max}$；约束 C5 表示语义压缩比 ρ_u 的范围条件；约束 C6 表示图像重建质量 PSNR_u 不低于重建图像质量的最低阈值 $\text{PSNR}_{\min}$。

2　算法设计

2.1　MDP 构建

编码器深度选择是离散问题，算力和带宽分配是连续问题，因此所提优化问题是一个混合整数非线性规划问题，难以用传统优化方法求解。本文将节能优化问题转化为 MDP 问题，使用 RL 算法求解。具体来说，将 BS 看作智能体，智能体在一个回合内完成所有终端的资源分配决策。一个回合包含 T 个决策步，定义决策步数 T 等于终端总数 U，即每个终端在一个回合中被服务一次，对应一个决策步。在每一个决策步 $t \in \{1,2,\cdots,T\}$，智能体为一个终端执行一次资源分配决策，并接收对应的奖励；当所有终端完成资源分配后回合结束。MDP 的关键要素定义如下。

（1）状态空间 $\mathcal{S}$。将系统资源状态建模为状态输入，包括 BS 的算力分配状态、终端占用链路带宽情况，以及终端的任务配置。为遵循最大最小公平性原则，所有终端按算力进行升序排序，构建终端服务队列 $\mathcal{Q} = [u_1, u_2, \cdots, u_T]$，其中 u_t 表示当前的终端服务索引，$u_t \in \{1,2,\cdots,U\}$。状态 $s_t \in \mathcal{S}$ 可表示为：

$$s_t = \left\{ u_t, \boldsymbol{M}, f_{\text{re}}^B, B_{\text{re}}, \{s_u\}_{u=1}^{U} \right\} \tag{17}$$

其中，$\boldsymbol{M} = [m_1, m_2, \cdots, m_U]$ 为服务掩码向量，用于标记已完成资源分配的终端，终端 u 已完成资源分配时 $m_u = 1$，终端 u 尚未被服务时 $m_u = 0$；f_{re}^B 表示 BS 剩余算力；B_{re} 为剩余带宽；$s_u = \{d_u, h_u, f_u^{\max}, \tau_{\max}\}$ 为终端 u 的任务特征，包括图像数据大小 d_u、信道增益 h_u、终端算力 $f_u^{\max}$、最大可容忍时延 $\tau_{\max}$。

（2）动作空间 $\mathcal{A}$。智能体对当前状态信息进行采样并选择动作 $a_t \in \mathcal{A}$：

$$a_t = \{l_t, f_t^B, B_t\} \tag{18}$$

其中，$l_t \in \{1,2,\cdots,L\}$ 表示 t 时刻终端选择的编码器深度，直接决定语义压缩比；f_t^B 表示 t 时刻终端使用的 BS 算力；B_t 表示 t 时刻终端占用的带宽。

（3）奖励函数 $\mathcal{R}$。以最小化系统总能耗为目标，同时需满足时延、重建质量等约束，因此奖励函数设计为：

$$r_t(s_t, a_t) = -E_t(s_t, a_t) - \mathcal{C}_{\text{penalty}} \tag{19}$$

其中，E_t 表示当前服务终端从 BS 编码、下行传输到终端解码的总能耗，$\mathcal{C}_{\text{penalty}}$ 表示违反约束的惩罚项，具体设置为：

$$\mathcal{C}_{\text{penalty}} = \lambda_1 \cdot \frac{\max\left(0, t_t^{\text{enc}} + t_t^{\text{trans}} + t_t^{\text{dec}} - \tau_{\max}\right)}{\tau_{\max}} + \lambda_2 \cdot \frac{\max\left(0, f_t^B - f_{\text{re}}^B\right)}{f_{\max}^B} + \lambda_3 \cdot \frac{\max\left(0, B_t - B_{\text{re}}\right)}{B} + \lambda_4 \cdot \frac{\max\left(0, \text{PSNR}_{\min} - \text{PSNR}_t\right)}{\text{PSNR}_{\min}} \tag{20}$$

其中，t_t^{enc}、t_t^{trans}和t_t^{dec}分别表示当前服务终端在BS编码、下行传输和终端解码的时延，PSNR_t表示当前服务终端的图像重建质量，λ_1、λ_2、λ_3和λ_4分别为时延、BS算力、链路带宽与图像重建质量约束的惩罚系数，用于平衡能耗与约束满足之间的关系。

2.2 稀疏注意力改进的PPO算法

针对本文构建问题中离散与连续变量并存的混合动作空间，本文在PPO框架下采用共享特征表示的多分支Actor-价值（Critic）结构，其中Actor网络由多个并行输出分支组成，分别对应带宽分配、基站算力分配以及语义编码器深度选择，不同分支用于建模连续与离散动作的决策过程，同时设置Critic网络对状态价值进行估计。其中，连续动作通过参数化概率分布进行采样，离散动作通过分类分布进行决策，并在策略更新过程中对各分量的对数概率进行统一聚合，从而在统一的策略优化目标下实现对混合动作的联合建模与协同决策。同时，为避免动作越界，本文对连续动作输出进行归一化映射，将其转换为满足总带宽与总算力约束的比例分配形式，从而保证所有资源分配结果始终位于可行域内；对于离散动作，则通过预定义动作集合进行索引映射，确保语义编码器深度选择的合法性。该设计避免了额外的动作裁剪操作，在保证约束满足的同时提高了策略学习的稳定性。

在多终端场景中，随着终端数量的增加，状态空间维度线性增长、无关信息冗余严重，导致PPO算法决策效率降低，收敛速度缓慢。基于此，本文在PPO算法的Actor网络中引入了稀疏注意力模块，通过仅保留与当前智能体决策相关性最强的k个终端的状态信息，过滤冗余信息，提高了算法的决策效率和收敛速度。

（1）状态特征嵌入

在状态输入阶段，引入状态特征嵌入机制，使模型能够感知终端和资源状态之间的多维空间关系，从而建立全局空间感知。状态嵌入层接收当前全局状态s_t。对于每个终端u，输出表示为：

$$\boldsymbol{e}_u^{(t)} = \text{Linear}\left(s_t\right) \tag{21}$$

其中，Linear(·)表示统一嵌入维度的线性投影层。最后形成嵌入特征矩阵$\boldsymbol{E}^{(t)} = \left[\boldsymbol{e}_1^{(t)}, \boldsymbol{e}_2^{(t)}, \cdots, \boldsymbol{e}_U^{(t)}\right]^{\text{T}}$，其中$\boldsymbol{e}_u^{(t)}$表示当前终端嵌入特征，随后将其用作注意力计算的Query输入。

（2）稀疏注意力计算

通过稀疏注意力模块计算注意力权重，从而获取最相关的特征。每个终端u嵌入向量$\boldsymbol{e}_u^{(t)}$经独立的线性映射后得到Key和Value投影。

$$\boldsymbol{k}_u^{(t)} = \boldsymbol{W}^K \cdot \boldsymbol{e}_u^{(t)} \tag{22}$$

$$\boldsymbol{v}_u^{(t)} = \boldsymbol{W}^V \cdot \boldsymbol{e}_u^{(t)} \tag{23}$$

其中，$\boldsymbol{W}^K$和$\boldsymbol{W}^V$为投影矩阵，$\boldsymbol{k}_u^{(t)}$和$\boldsymbol{v}_u^{(t)}$分别为Key向量和Value向量。因此，全局Key和Value矩阵表示为：

$$\boldsymbol{K}^{s(t)} = \left[\boldsymbol{k}_1^{(t)}, \boldsymbol{k}_2^{(t)}, \cdots, \boldsymbol{k}_u^{(t)}\right] \tag{24}$$

$$\boldsymbol{V}^{s(t)} = \left[\boldsymbol{v}_1^{(t)}, \boldsymbol{v}_2^{(t)}, \cdots, \boldsymbol{v}_U^{(t)}\right] \tag{25}$$

$\boldsymbol{e}_u^{(t)}$经线性变换后得到终端u的Query向量$\boldsymbol{q}_u^{(t)}$，通过计算$\boldsymbol{q}_u^{(t)}$与Key矩阵$\boldsymbol{K}$的相关性，可以计算出每个节点的注意力分数，并引入掩码向量屏蔽已服务终端。

$$\alpha_u = \begin{cases} \dfrac{\boldsymbol{q}_u^{(t)} \boldsymbol{K}^{\text{T}}}{\sqrt{d_k}} & , m_u = 0 \\ 0 & , m_u = 1 \end{cases} \tag{26}$$

其中，d_k是Key向量的维度。

注意力分数向量$\boldsymbol{\alpha} = \left[\alpha_1, \alpha_2, \cdots, \alpha_U\right]$经Top-k Softmax函数激活权重最高的$k$个终端，同时将其他权重置0，得到稀疏注意力权重为：

$$\boldsymbol{\alpha}' = \text{SparseAtten}\left(\boldsymbol{\alpha}\right) \tag{27}$$

$\boldsymbol{\alpha}'$与$\boldsymbol{V}^{s(t)}$加权聚合得到决策节点的稀疏特征表示为：

$$\boldsymbol{e}_s = \boldsymbol{\alpha}' \cdot \boldsymbol{V}^{s(t)} = \sum_{u=1}^{U} \alpha_u' \boldsymbol{v}_u^{(t)} \tag{28}$$

得到的特征向量$\boldsymbol{e}_s$集成与当前终端强相关的终端特征，同时抑制冗余或不相关的终端特征，从而为Actor网络提供轻量级状态输入。

最后，将特征$\boldsymbol{e}_s$输入Actor网络中，得到动作概率分布：

$$\pi_{\theta_a}(a_t|s_t) = \mathrm{MLP}\left(\boldsymbol{e}_s\right) \tag{29}$$

（3）稀疏注意力驱动的 Critic 网络

在价值估计阶段，为了捕捉关键节点之间的全局依赖关系，本文设计了一种稀疏注意力驱动的 Critic 网络，从而实现高效准确的状态价值估计。具体而言，采用池化方法将稀疏注意力过滤后的所有关键节点特征聚合成单个全局特征向量 $\boldsymbol{e}_u^{\mathrm{all}}$，然后通过 MLP 网络回归并获得状态价值估计，输出状态价值函数 $V\left(s_t\right)$。

$$V\left(s_t\right) = \mathrm{MLP}\left(\boldsymbol{e}_u^{\mathrm{all}}\right) = \mathrm{MLP}\left(\mathrm{Pool}\left(\mathrm{SparseAtten}\left(s_t\right)\right)\right) \tag{30}$$

上述改进确保了 PPO 算法在高维状态空间中更好的泛化能力和训练稳定性。本文提出的基于轻量化 Swin Transformer 的资源自适应语义压缩策略的伪代码如算法 1 所示。

算法 1　基于轻量化 Swin Transformer 的资源自适应语义压缩策略

输入　终端数量 U、带宽 B、基站算力 f_B、终端算力 $f_u^{\max}$、原始数据量 d_u 等环境参数，以及学习率 lr、折扣因子 γ 等 PPO 超参数

输出　最优策略 π^*、系统总能耗 E_{total}

初始化环境参数，根据终端算力 f_u^B 对终端进行升序排序，形成终端服务队列 $\mathcal{Q}$
初始化 Actor 网络参数 θ、Critic 网络参数 ϕ，经验回放缓冲区 $\mathcal{D} \leftarrow \varnothing$
for episode = 1 to max_episodes do
　获取初始状态 s_0
　for t = 1 to T do
　　从队列 $\mathcal{Q}$ 中获取当前的终端服务索引 u_t
　　根据式(21)得到状态嵌入特征矩阵 $\boldsymbol{E}^{(t)} = \left[\boldsymbol{e}_1^{(t)}, \boldsymbol{e}_2^{(t)}, \cdots, \boldsymbol{e}_U^{(t)}\right]^{\mathrm{T}}$
　　根据式(22)~式(28)经稀疏注意力计算得到稀疏状态特征 $\boldsymbol{e}_s$
　　根据 Actor 网络 $\pi_{\theta_a}(a_t|s_t) = \mathrm{MLP}\left(\boldsymbol{e}_s\right)$ 采样并执行动作 $a_t = \left\{l_t, f_t^B, B_t\right\}$
　　根据式(7)~式(13)计算 E_t，并根据式(19)计算 r_t
　　选择下一个服务终端 u_{t+1} 并更新下一状态 s_{t+1}
　　存储样本 $\left(s_t, a_t, r_t, s_{t+1}\right)$ 至 $\mathcal{D}$
　end for
　计算回报 $R_t = \sum_{i=t}^{T} \gamma^{i-t} r_i$ 和优势函数 $\hat{A}_t$
　for epoch = 1 to max_epochs do
　　从 $\mathcal{D}$ 中随机采样小批量数据
　　Critic 网络通过式(30)计算 $V\left(s_t\right)$
　　基于 PPO 算法裁剪目标函数，使用采样得到的小批量数据，更新 θ 和 ϕ
　end for
　更新旧策略参数 $\theta_{\mathrm{old}} \leftarrow \theta$
　清空经验回放缓冲区 $\mathcal{D}$
end for
$E_{\mathrm{total}} \leftarrow \sum_{t=1}^{T} E_t$
返回优化后的策略 $\pi^* = \pi_\theta$、系统总能耗 E_{total}

3　仿真结果与性能评估

3.1　仿真参数设置

本节在 Python 环境和 Visual Studio Code 平台上进行仿真分析。考虑一个半径为 500 m 的圆形网络，BS 位于网络中心，U 个终端随机分布在网络中。信道增益包括路径损耗和阴影效应，路径损耗模型定义为 128.1 + 37.6lg$d$（其中 d 以 km 为单位），阴影效应因子为 6 dB，每个终端传输 1 200 MB 的原始图像数据（d_u = 1 200 MB,$\forall u \in \mathcal{U}$），任务完成期限为 600 s（$\tau_{\max}$ = 600 s），总系统带宽为 20 MHz，噪声功率谱密度为−174 dBm/Hz，BS 的发射功率为 20 dBm，终端最大本地计算频率均匀分布在 0.62~1.28 GHz。终端每 GPU 周期可执行的浮点运算次数为 ε_u=1 024，BS 每 GPU 周期可执行的浮点运算次数为 ε_B=13 824，用户可容忍的最低图像重建质量为 25 dB，有效开关电容系数 κ_U=κ_B=10^{-28}[24]。

本文策略的训练参数配置如表 1 所示。折扣因子 γ 设为 0.99，使智能体在决策中重视长期回报；广义优势估计系数 λ 设为 0.95，以稳定优势函数估计；PPO 裁剪系数 ε 设为 0.1，用于稳定更新；价值函数损失系数设为 0.5，以平衡策略与价值优化。Actor 与 Critic 网络共享特征提取层，隐藏层维度设为 256。学习率 1×10^{-4} 通过收敛性实验确定为收敛速度与稳定性最优的值。总训练回合数设为 5 500，

以保证所有学习率下奖励均趋于稳定。经验回放缓冲区容量设为2 048，每轮随机采样64条样本进行3次更新，平衡了样本效率与计算代价。熵正则化系数0.01鼓励探索，避免策略过早陷入局部最优。稀疏注意力Top-k值取3，在过滤无关状态的同时保留与当前决策最相关的终端特征，提高了高维状态下的决策效率。算法在PyTorch框架下实现，采用Adam优化器进行训练。

表1 训练参数配置

训练参数	参数值
学习率lr	1×10^{-4}
折扣因子γ	0.99
广义优势估计系数λ	0.95
PPO裁剪系数ε	0.1
价值函数损失系数	0.5
熵正则化系数	0.01
总训练回合数max_episodes	5 500
小批量大小mini_batch	64
更新轮次max_epochs	3
网络隐藏层维度	256
缓冲区$\mathcal{D}$大小	2 048
稀疏注意力Top-k值	3

3.2 性能分析

3.2.1 改进的语义编码器结构的性能分析

（1）稀疏注意力机制的超参数敏感性分析

本节对所提“全局压缩+Top-k选择”稀疏注意力机制中的关键超参数压缩块数量M与Top-k选择数量k进行敏感性分析，旨在揭示模型性能与计算开销之间的权衡关系，并确定后续实验的最优参数配置。表2展示了不同参数组合下，轻量化语义编码器在编码整个Kodak数据集时的重建质量和计算量（以FLOPs为单位）。

从表2可以看出，与标准Swin-T相比，引入稀疏注意力机制的编码器均获得了不同程度的计算开销下降。当压缩块数量M固定时，随着k的减小，模型的PSNR和FLOPs均呈现下降趋势。这表明减小k会牺牲部分重建精度以换取更低的计算开销。同时，在相同k值下，减小M同样降低了PSNR和FLOPs，表明过少的压缩块会导致全局上下文信息的丢失。以标准Swin-T为基准，对比3种M取值下$k=2$的表现：$M=8$时，PSNR降幅为7.1%，FLOPs降幅为8.8%；$M=7$时，PSNR降幅为6.9%，FLOPs降幅为8.5%；$M=4$时，PSNR降幅为9.8%，FLOPs降幅为9.4%。对比可见，$M=7$与$M=8$在FLOPs降幅上仅相差0.3%，PSNR损失基本持平，这表明将M从8微调至7几乎不牺牲质量即可获得相近的轻量化收益。然而，当M进一步降至4时，PSNR降幅显著扩大至9.8%，逼近30 dB以下的感知质量门槛，边际损失明显加剧。在$M=7$的基础上，对比参数k的影响：$k=3$时，PSNR降幅为5.4%，FLOPs降幅仅3.2%，计算收益有限；$k=1$时，FLOPs降幅为15.8%，PSNR降幅扩大至8.8%，质量牺牲较大。相比之下，$k=2$以8.5%的FLOPs降幅实现了仅6.9%的PSNR损失，在保证重建质量与节省计算开销之间取得了最优均衡。综上，后续实验选取$M=7, k=2$作为稀疏注意力机制的参数配置。

表2 不同参数设置下的语义编码器性能

参数M,k取值	平均PSNR/dB	计算量/FLOPs
标准Swin-T	33.05	848.46
$M=8, k=3$	31.28	825.51
$M=8, k=2$	30.69	773.38
$M=8, k=1$	29.92	713.25
$M=7, k=3$	31.28	820.99
$M=7, k=2$	30.77	775.98
$M=7, k=1$	30.14	713.98
$M=4, k=3$	30.81	816.37
$M=4, k=2$	29.80	769.06
$M=4, k=1$	28.75	720.75

（2）稀疏注意力机制改进前后的编码器性能分析

表3对比了不同编码器深度下，原始Swin Transformer与稀疏注意力机制改进的Swin Transformer在编码整个Kodak数据集时的重建质量和计算量（以FLOPs为单位）。从表3可以看出，稀疏注意力机制改进后Swin Transformer的计算量小于原始Swin Transformer，这种改进是以图像重建质量的下降为代价的，但图像重建质量仍在可接受范围内。图4为重建图像的视觉比较。

表 3　稀疏注意力机制改进前后的语义编码器性能比较

语义编码器	编码器深度	平均 PSNR/dB	计算量/FLOPs
原始 Swin Transformer	[2, 2, 2, 2]	32.02	532.58
	[2, 2, 6, 2]	33.05	847.46
	[2, 2, 18, 2]	33.97	1 636.24
稀疏注意力机制改进的 Swin Transformer	[2, 2, 2, 2]	29.66	480.24
	[2, 2, 6, 2]	30.77	775.98
	[2, 2, 18, 2]	31.69	1 497.26

（3）不同编码器深度下的语义压缩性能分析

表 4 对比了语义编码器在不同深度下编码整个 Kodak 数据集时的语义压缩比、图像重建质量和计算量（以 FLOPs 为单位）。从表 4 可以看出，编码器深度与语义压缩比呈正相关，计算复杂度随深度近似线性增长，图像重建质量在较浅深度配置下提升较为明显，在较深深度配置下增益逐渐放缓，呈现典型的边际收益递减特性。这表明更深的编码器能够提取更精炼的语义信息，以更少的数据量表征原始内容，相应地消耗更多的算力，图像重建质量有所提升。上述结果验证了不同编码器深度下语义压缩比与算力资源消耗之间的权衡关系，从而印证了本文利用门控网络动态调节编码器深度、为终端定制差异化语义压缩比这一设计思想的合理性。

表 4　不同深度下语义编码器性能比较

编码器深度	语义压缩比	平均 PSNR/dB	计算量/FLOPs
[2, 2, 2, 2]	2.272	29.66	480.24
[2, 2, 4, 2]	2.667	30.11	610.32
[2, 2, 6, 2]	3.840	30.77	775.98
[2, 2, 8, 2]	4.741	31.05	920.18
[2, 2, 10, 2]	5.988	31.19	1 085.63
[2, 2, 14, 2]	8.696	31.47	1 280.05
[2, 2, 18, 2]	12.048	31.69	1 497.26

3.2.2　本文策略的性能分析

通过实验仿真对本文策略性能进行验证，包括收敛性能和节能性能。为全面评估本文策略的有效性，本文将其与以下策略进行对比分析。

均匀资源分配（uniform allocation, UA）策略：该策略平均分配 BS 的总算力和系统总带宽给所有终端，同时所有终端采用统一的、固定的语义压缩比。

贪婪启发式分配（greedy heuristic allocation, GHA）策略：该策略将终端按本地算力从低到高排序，优先服务算力最低、解码能力最受限的终端。若当前压缩比下无法满足约束，则尝试次高的压缩比，直至找到可行解。在为该终端分配资源后，更新系统的剩余带宽与 BS 剩余算力，再以相同规则处理下一个终端。

图 4　重建图像的视觉比较

基于 PPO 算法的固定语义压缩比的资源分配（PPO-based resource allocation with fixed semantic compression ratio, PPO-FSCR）策略：该策略中终端采用统一的语义压缩比，利用 PPO 算法对 BS 算力和带宽分配进行优化。

基于深度确定性策略梯度算法的资源自适应语义压缩（DDPG-based resource-adaptive semantic compression, DDPG-RASC）策略：该策略根据终端算力进行语义压缩比、BS 算力和带宽分配，给高算力终端分配高语义压缩比、高 BS 算力和低带宽，给低算力终端分配低语义压缩比、低 BS 算力和高带宽，以在有限算力和带宽下获得高性能。采用 DDPG 算法对语义压缩比、BS 算力和带宽分配进行优化。

基于 PPO 算法的资源自适应语义压缩（PPO-based resource-adaptive semantic compression, PPO-RASC）策略：该策略采用 PPO 算法对语义压缩比、BS 算力和带宽分配进行优化。

（1）收敛性分析

图 5 展示了 PPO 算法和本文策略分别在不同学习率下的收敛性能，旨在验证本文策略在优化问题式(15)中的训练效率与稳定性，曲线经过萨维茨基–戈莱（Savitzky-Golay）滤波器（窗口长度为 50）平滑处理，以更清晰地展示训练趋势。从图 5 可以看出，随着迭代次数的增加，所有学习率下的奖励值均呈上升趋势，且最终均达到相近的稳定奖励水平，表明两种算法在本优化问题上均具有良好的稳定性与鲁棒性。同时，在相同学习率条件下，本文策略整体表现出更快的收敛速度。在训练初期，本文策略的奖励值提升更迅速，并且能够在较少的训练回合内接近最终稳定奖励水平，PPO 算法则需要更多的训练迭代才能达到类似的性能。这是因为稀疏注意力机制能够使智能体在训练过程中更快地捕获影响系统性能的关键状态信息，从而加快策略学习过程。

（2）节能分析

图 6 对比了不同策略下终端数量变化对系统总能耗的影响。从图 6 可以观察到，随着终端数量的增多，所有策略的系统总能耗均呈增长趋势。这是因为更多的终端意味着更高的语义编解码开销和带宽需求，系统所需的算力和带宽总需求越大。UA 由于完全不考虑终端算力差异和资源复用，能耗最高。GHA 通过简单的规则匹配了终端算力和压缩比，性能优于 UA，但其贪婪的决策方式缺乏全局视野，容易陷入局部最优。与 UA 和 GHA 相比，基于 RL 的策略能通过学习历史经验，做出更具前瞻性的资源分配决策，因此能耗普遍更低。其中，PPO-FSCR 的能耗始终最高且增长最快，其原因在于该策略采用固定语义压缩比。DDPG-RASC 和 PPO-RASC 通过自适应压缩比改善了压缩效率，能耗优于 PPO-FSCR。与上述方法相比，本文策略通过在 Actor 网络中引入稀疏注意力模块，可以更有效地捕获多终端状态信息之间的关键关联关系，使智能体能够更合理地联合优化语义压缩比、BS 算力和带宽资源。综上，本文策略的系统总能耗始终是最低的，且增长平缓，证明了本文策略在多终端场景下具有良好的可扩展性。

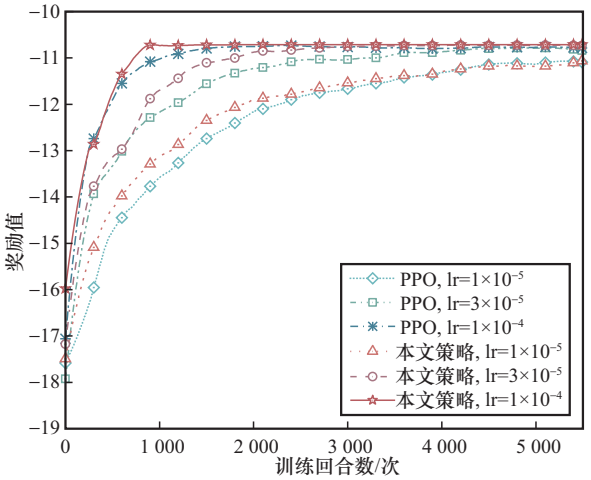


图 5　不同学习率下 PPO 算法与本文策略的收敛性能对比

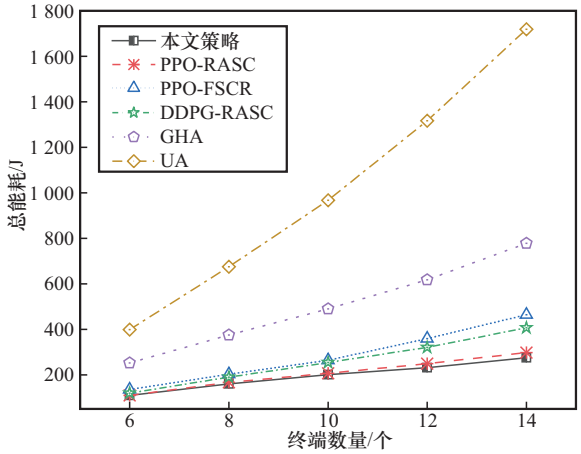


图 6　终端数量对系统总能耗的影响

图 7 展示了系统总能耗随原始数据量增大的变化趋势。随着数据量增大，语义编解码开销与传输时延均会显著增加，因而所有策略的能耗均随原始数据量上升。同样地，UA 的系统总能耗最高，GHA 的系统总能耗仅次于 UA，4 种 RL 策略的能耗均优于 UA 和 GHA。其中，PPO-FSCR 对大数据量最不适应，其固定压缩比会导致传输负担急剧增加。DDPG-RASC 和 PPO-RASC 通过自适应调整语义压缩比，在一定程度上缓解了数据量增长带来的能耗压力，其整体性能优于 PPO-FSCR。本文策略在所有数据量下均能保持最优能耗表现，且能耗提升幅度最小，这得益于其通过全局优化压缩比与资源分配，使数据量扩大带来的负面影响得到最小化。上述结果验证了本文策略在处理大规模数据传输任务时的优越性。

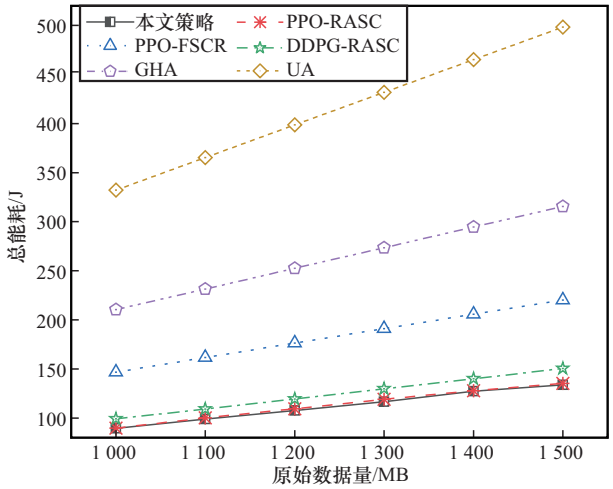


图 7　原始数据量对系统总能耗的影响

图 8 为带宽对系统总能耗的影响。当带宽增加时，数据传输速率提升、传输时延降低，因此所有策略的能耗均呈下降趋势。其中，PPO-FSCR 的能耗最高且下降幅度最大，这是因为在低带宽条件下，固定语义压缩比导致语义数据量较大，传输效率极低，高带宽能够显著缓解其传输瓶颈，因此呈现出较大幅度的能耗改善。同样地，UA 的系统总能耗最高，GHA 次之，4 种 RL 策略的能耗普遍低于 UA 和 GHA。其中，DDPG-RASC 和 PPO-RASC 利用自适应语义压缩比能够在低带宽条件下主动提高压缩比，减轻传输负担，因此在所有带宽条件下性能均优于 PPO-FSCR。本文策略在所有带宽条件下均保持最低能耗，特别是在低带宽情况下，通过自适应提高语义压缩比并合理分配算力和带宽，显著缓解了带宽受限所带来的能耗瓶颈。上述结果验证了本文策略在不同带宽条件下的稳定性和优越的能效性能。

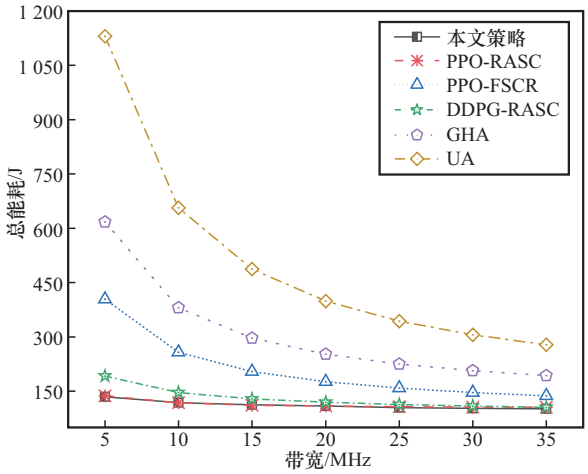


图 8　带宽对系统总能耗的影响

3.3　消融实验分析

为验证本文策略中各核心组件的有效性，本节设计了系统的消融实验，对比了以下策略。

本文策略：采用引入门控网络和稀疏注意力的轻量化语义编码器，搭配稀疏注意力改进的 PPO 算法。

门控网络和稀疏注意力（gating and sparse attention, GS）策略：采用所提轻量化语义编码器，搭配标准 PPO 算法。

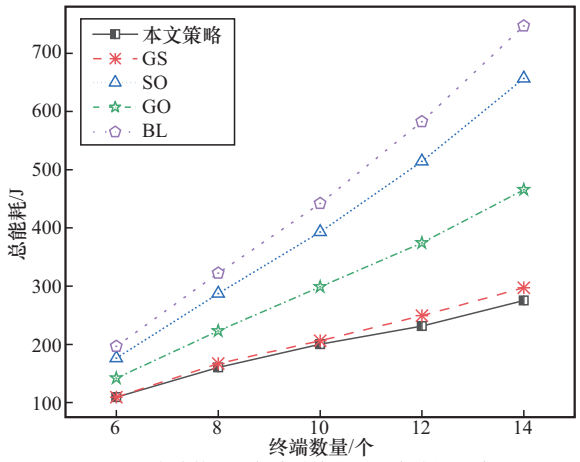
仅稀疏注意力（sparse attention only, SO）策略：采用固定深度但引入稀疏注意力的语义编码器，搭配标准 PPO 算法。

仅门控网络（gating only, GO）策略：采用引入门控网络的可变深度语义编码器，搭配标准 PPO 算法。

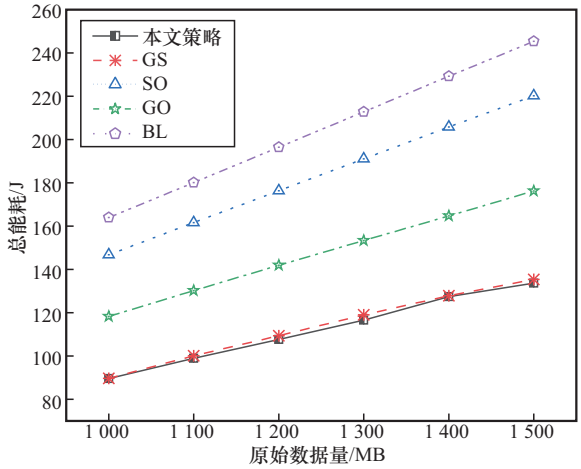
基线（baseline, BL）策略：采用固定深度的语义编码器与标准 PPO 算法。

图 9 展示了消融实验的结果。对比 BL 与 GO 方案发现，系统总能耗显著下降，表明门控网络通过为终端定制差异化压缩比，能够有效适配终端算力差异，提高了算力、带宽资源的利用率。对比 BL 与 SO 方案发现，系统总能耗也有所降低，说明语义编码器中的稀疏注意力机制通过减少计算开销，获得了能效增益。GS 结合了门控网络与稀疏注意力机制两者的优势，使系统总能

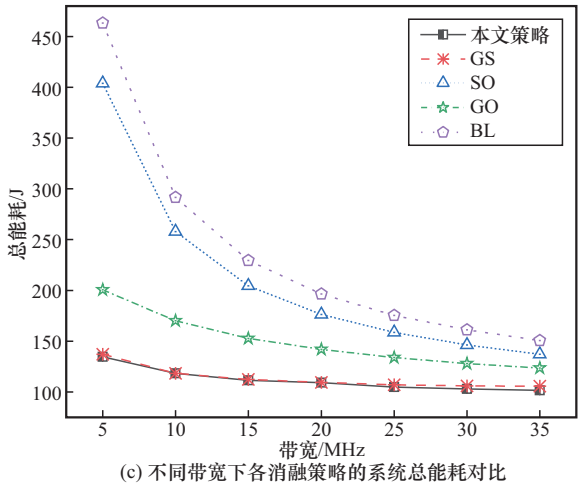
耗进一步降低。由图9(a)可知，本文策略在GS方案的基础上引入了稀疏注意力改进的PPO算法，使智能体在高维状态空间中具备更高效的决策能力，能够制定更优的资源分配策略，从而实现了最低的系统总能耗。上述结果表明，本文策略中的各核心组件均对整体性能提升起到了积极作用。



(a) 不同终端数量下各消融策略的系统总能耗对比



(b) 不同原始数据量下各消融策略的系统总能耗对比



(c) 不同带宽下各消融策略的系统总能耗对比

图9　不同参数配置下各消融策略的系统总能耗对比

4　结束语

本文提出了一种基于轻量化Swin Transformer的资源自适应语义压缩策略，旨在解决多终端下行语义通信场景中因终端算力差异导致的资源利用率低的问题。该策略通过引入门控网络实现动态可调深度的语义编码器，并用“全局压缩+Top-k选择”稀疏注意力机制改进了Swin Transformer，降低了语义编码的计算开销。将优化问题转化为MDP，通过PPO算法实现高效求解，并通过引入稀疏注意力机制，提高了PPO算法的训练效率。仿真结果表明，本文策略能在保证图像重建质量的前提下减少计算开销，同时在节能性能方面优于其他策略。

参考文献：

[1] Gündüz D, Qin Z J, Aguerri I E, et al. Beyond transmitting bits: context, semantics, and task-oriented communications[J]. IEEE Journal on Selected Areas in Communications, 2023, 41(1): 5-41.

[2] Bourtsoulatze E, Burth Kurka D, Gündüz D. Deep joint source-channel coding for wireless image transmission[J]. IEEE Transactions on Cognitive Communications and Networking, 2019, 5(3): 567-579.

[3] Yang K, Wang S X, Dai J C, et al. SwinJSCC: taming swin transformer for deep joint source-channel coding[J]. IEEE Transactions on Cognitive Communications and Networking, 2025, 11(1): 90-104.

[4] 香晏, 赵响, 黄军韬. 一种基于残差连接的Swin Transformer增强型联合编码架构设计[J]. 无线电工程, 2025, 55(5): 905-912.
Xiang Y, Zhao X, Huang J T. A residual connection-based swin transformer enhanced joint encoding architecture design[J]. Radio Engineering, 2025, 55(5): 905-912.

[5] 张平, 牛凯, 姚圣时, 等. 面向未来的语义通信: 基本原理与实现方法[J]. 通信学报, 2023, 44(5): 1-14.
Zhang P, Niu K, Yao S S, et al. Semantic communications for future: basic principle and implementation methodology[J]. Journal on Communications, 2023, 44(5): 1-14.

[6] Tang Q, Zhang B W, Liu J J, et al. Dynamic token pruning in plain vision transformers for semantic segmentation[C]//Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2023: 777-786.

[7] Pan J T, Bulat A, Tan F W, et al. EdgeViTs: competing light-weight CNNs onMobile devices withVision transformers[C]//Computer Vision-ECCV 2022. Cham: Springer, 2022: 294-311.

[8] Gao Z T, Tong Z, Wang L M, et al. SparseFormer: sparse visual recognition via limited latent tokens[PP]. V1. (2023-04-07) [2026-04-26]. arXiv: arXiv.2304.03768.

[9] Yuan J Y, Gao H Z, Dai D M, et al. Native sparse attention: hardware-aligned and natively trainable sparse attention[C]//Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg: ACL, 2025: 23078-23097.

[10] Zhao Z X, Yang Z H, Gan X, et al. A joint communication and computation design for semantic wireless communication with probability graph[J]. Journal of the Franklin Institute, 2024, 361(13): 107055.

[11] Yang S H, Shen B, Huang X G. Optimizing semantic spectral efficiency in wireless image transmission: a PPO-driven resource allocation scheme[J]. IEEE Communications Letters, 2025, 29(6): 1466-1470.

[12] Cang Y H, Chen M, Yang Z H, et al. Online resource allocation for semantic-aware edge computing systems[J]. IEEE Internet of Things Journal, 2024, 11(17): 28094-28110.
[13] Zhao S F, Liu T T, Jin H, et al. SPViT: accelerate vision transformer inference on mobile devices *via* adaptive splitting and offloading[J]. IEEE Transactions on Mobile Computing, 2025, 24(10): 9303-9318.
[14] Ji Z L, Qin Z J, Tao X M, et al. Resource optimization for semantic-aware networks with task offloading[J]. IEEE Transactions on Wireless Communications, 2024, 23(9): 12284-12296.
[15] 顾健华, 冯建华, 许辉阳, 等. 基于有向图与卷积网络强化学习的端侧协同算力资源分配方法[J]. 电子学报, 2025, 53(6): 1771-1783.
Gu J H, Feng J H, Xu H Y, et al. Directed graph and convolutional network reinforcement learning for terminal-side collaborative computing resource allocation scheme[J]. Acta Electronica Sinica, 2025, 53(6): 1771-1783.
[16] Nguyen L X, Tun Y L, Tun Y K, et al. Swin transformer-based dynamic semantic communication for multi-user with different computing capacity[J]. IEEE Transactions on Vehicular Technology, 2024, 73 (6): 8957-8972.
[17] Zhang J, Zhang Y, Ji B F, et al. Feature-driven semantic communication for efficient image transmission[J]. Entropy, 2025, 27(4): 369.
[18] Albaseer A, Abdallah M. Tailoring semantic communication at network edge: a novel approach using dynamic knowledge distillation[C]// Proceedings of the ICC 2024 - IEEE International Conference on Communications. Piscataway: IEEE Press, 2024: 1455-1460.
[19] Zheng Y P, Zhang T K, Mu X D, et al. Joint semantic transmission and resource allocation for intelligent computation task offloading in MEC systems[J]. IEEE Transactions on Wireless Communications, 2025, 24 (10): 8756-8770.
[20] Liu C H, Guo C L, Yang Y, et al. Adaptable semantic compression and resource allocation for task-oriented communications[J]. IEEE Transactions on Cognitive Communications and Networking, 2024, 10(3): 769-782.
[21] Zheng G Y, Wen M W, Ning Z L, et al. Computation-aware offloading for DNN inference tasks in semantic communication assisted MEC systems[J]. IEEE Transactions on Wireless Communications, 2025, 24(4): 2693-2706.
[22] Zheng Y P, Zhang T K, Huang R, et al. Joint computing offloading and resource allocation for classification intelligence tasks in MEC systems[J]. IEEE Transactions on Network and Service Management, 2026, 23: 1086-1099.
[23] Chen B, Huang Y J, Qiu H, et al. Image compression for resource-constrained AIoT system with compressed sensing[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2025, 55(8): 5464-5476.
[24] Hu H, Song K F, Fan R F, et al. Energy-efficient image semantic communication: architecture design and optimal joint allocation of communication and computation resources[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2025, 35(7): 6466-6480.

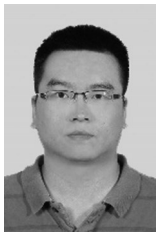
[作者简介]

郑飞（1982– ），男，博士，桂林电子科技大学副教授、硕士生导师，主要研究方向为卫星通信网、电力通信网、资源管理、移动边缘计算、确定性传输、路由规划及协议、AI 大模型算法应用。

吴东颖（2002– ），女，桂林电子科技大学硕士生，主要研究方向为语义通信、资源管理、AI 大模型算法应用。

李世超（1986– ），男，博士，桂林电子科技大学副教授、硕士生导师，主要研究方向为 6G 无线通信、车联网、移动边缘计算、空天地一体化网络、深度强化学习。

仇洪冰（1963– ），男，博士，桂林电子科技大学教授、博士生导师，主要研究方向为宽带无线通信、通信感知一体化、通信信号处理、通信网络。



邵苏杰（1985– ），男，博士，北京邮电大学副教授、博士生导师，主要研究方向为网络管理、能源互联网信息通信、边缘计算。

喻鹏（1986– ），男，博士，北京邮电大学副教授、博士生导师，主要研究方向为 B5G/6G 网络智能管控。

丰雷（1987– ），男，博士，北京邮电大学副教授、博士生导师，主要研究方向为网络智能管理、能源互联网信息通信。

赵继龙（2001– ），男，桂林电子科技大学硕士生，主要研究方向为卫星通信网络资源管理、边缘计算、强化学习和人工智能。