Appendix

Peng Chen[1] and Jixian Zhang[1,2(✉)]

[1] School of Information Science and Engineering, Yunan University
[2] Yunnan Key Laboratory of Intelligent Systems and Computing, Yunan University
pengchen@stu.ynu.edu.cn, zhangjixian@ynu.edu.cn

Appendix-1

Semantic Information Shunt Transmission Mode

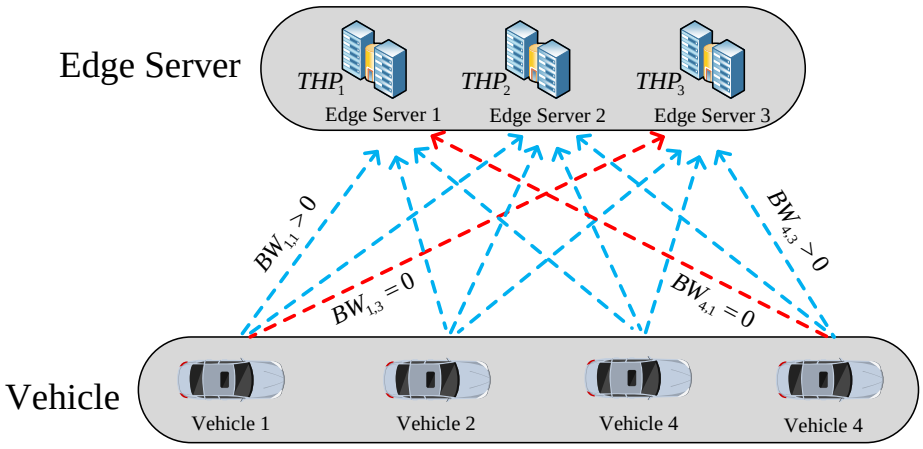

Fig. 1. Semantic information shunt transmission mode.

In the semantic communication-based IoV crowdsensing tasks, we consider a more flexible and efficient data shunt transmission model, as shown in 1, which allows users to offload data to multiple edge servers with different bandwidths. This shunt transmission mode can greatly increase the data transmission rate.

The Training Algorithm for SVRANet

We provide the training algorithm of SVRANet:

Algorithm 1 represents the training algorithm of SVRANet, where line 1 initializes the parameters (the learning rate, Lagrange multipliers, Lagrange multiplier period, etc.), α denotes the learning rate for the gradient ascent process that maximizes the user misreporting, and β denotes the learning rate for the gradient descent process that minimizes the loss function. When training SVRANet for T epochs, each epoch uses a minibatch dataset $S_t \subset S$, and 4-7 lines are subjected to gradient ascent to find the users' optimal misreporting $rgt_i^{(l)}(w^t)$. Then, lines 8-9 execute gradient descent to reduce the loss function and update the network parameters. Finally, the Lagrange multipliers are updated.

Algorithm 1 The training algorithm for SVRANet

Input: Minibatches $S_1, S_2, ..., S_T$, $S_t \subset S$, $t \in [0, T]$, Budget B

Initialize: $\forall \rho_\lambda > 0, \rho_\mu > 0, \rho_\eta > 0, \rho_\delta > 0, \rho_\gamma > 0,$
$T \in \mathbb{N}, R \in \mathbb{N}, T_\lambda \in \mathbb{N}, T_\mu \in \mathbb{N}, T_\eta \in \mathbb{N}, T_\delta \in \mathbb{N}, T_\gamma \in \mathbb{N}$
$w^0 \in \mathbb{R}^d, \lambda^0 \in \mathbb{R}^N, \mu^0 \in \mathcal{R}^N, \eta^0 \in \mathbb{R}, \delta^0 \in \mathbb{R}^{N \times M}, \gamma^0 \in \mathbb{R}^M.$

for $t = 0$ to T **do**
  Receive minibatch S_t
  Initialize false reports $\boldsymbol{b}_i^{(l)} \in F_i, \forall l \in S_t, i \in N$
  for $r = 0$ to R **do**
    $\forall l \in S_t, i \in \mathcal{N} : \boldsymbol{b}_i^{(l)} \leftarrow \boldsymbol{b}_i^{(l)} + \alpha \nabla_{\boldsymbol{b}_i^{(l)}} u_i^w \left((\boldsymbol{b}_{-i}, b_i) \right)$
  end for
  Compute $rgt\,(w)$, $irp\,(w)$, $bfp\,(w)$, $bwp\,(w)$, the gradient $thpp\,(w)$ and the gradient update w^t.
  $w^{t+1} \leftarrow w^t - \beta \nabla_w \mathcal{L}_{\rho_\lambda, \rho_\mu, \rho_\eta, \rho_\delta, \rho_\gamma} \left(w^t, \lambda^t, \mu^t, \eta^t, \delta^t, \gamma^t \right)$
  Update the Lagrange multipliers once in several iterations:
  if t is a multiple of T_λ **then**
    $\lambda_i^{t+1} \leftarrow \lambda_i^t + \rho_\lambda rgt_i \left(w^{t+1} \right), \forall i \in \mathcal{N}$
  else
    $\lambda_i^{t+1} \leftarrow \lambda_i^t$
  end if
  if t is a multiple of T_μ **then**
    $\mu_i^{t+1} \leftarrow \mu_i^t + \rho_\mu irp_i \left(w^{t+1} \right), \forall i \in \mathcal{N}$
  else
    $\mu_i^{t+1} \leftarrow \mu_i^t$
  end if
  if t is a multiple of T_η **then**
    $\eta^{t+1} \leftarrow \eta^t + \rho_\eta bf \left(w^{t+1} \right)$
  else
    $\eta^{t+1} \leftarrow \eta^t$
  end if
  if t is a multiple of T_δ **then**
    $\delta_{i,j}^{t+1} \leftarrow \delta_{i,j}^t + \rho_\delta bwp_{i,j} \left(w^{t+1} \right), \forall i \in \mathcal{N},\ j \in \mathcal{M}$
  else
    $\delta_{i,j}^{t+1} \leftarrow \delta_{i,j}^t$
  end if
  if t is a multiple of T_γ **then**
    $\gamma_j^{t+1} \leftarrow \gamma_j^t + \rho_\gamma thpp_j \left(w^{t+1} \right), \forall j \in \mathcal{M}$
  else
    $\gamma_j^{t+1} \leftarrow \gamma_j^t$
  end if
end for

Appendix-2

The Architecture of SVRANet

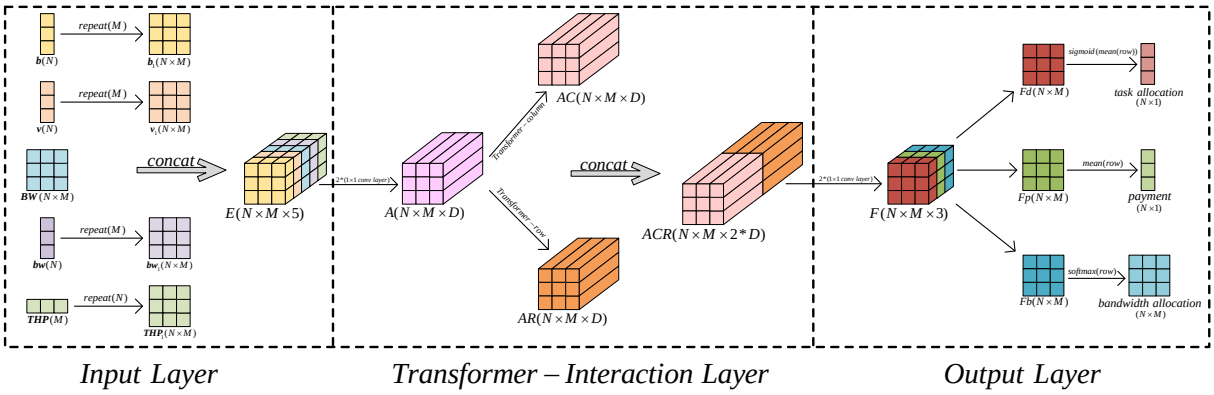

Fig. 2. The architecture of SVRANet.

SVRANet consists of 3 parts: $Input\ Layer$, $Transformer-Interaction\ Layer$ and $Output\ Layer$. The detailed descriptions of these 3 parts are as follows.

Input Layer The inputs of the Input Layer are bids $\boldsymbol{b} \in \mathcal{R}_{\geq 0}^{N}$, values $\boldsymbol{v} \in \mathcal{R}_{\geq 0}^{N}$, bandwidth requirements $\boldsymbol{bw} \in \mathcal{R}_{\geq 0}^{N}$, channel capacities $\boldsymbol{BW} \in \mathcal{R}_{\geq 0}^{N \times M}$, and edge server throughput levels $\boldsymbol{THP} \in \mathcal{R}_{\geq 0}^{M}$. We extend the sizes of $\boldsymbol{b}$, $\boldsymbol{v}$, $\boldsymbol{bw}$ and $\boldsymbol{THP}$ to $N \times M$, obtaining $\boldsymbol{b}_1 \in \mathcal{R}_{\geq 0}^{N \times M}$, $\boldsymbol{v}_1 \in \mathcal{R}_{\geq 0}^{N \times M}$, $\boldsymbol{bw}_1 \in \mathcal{R}_{\geq 0}^{N \times M}$ and $\boldsymbol{THP}_1 \in \mathcal{R}_{\geq 0}^{N \times M}$:

$$\boldsymbol{b}_1 = repeat - column(\boldsymbol{b}, M) \in \mathcal{R}_{\geq 0}^{N \times M}, \quad \boldsymbol{v}_1 = repeat - column(\boldsymbol{v}, M) \in \mathcal{R}_{\geq 0}^{N \times M}$$
$$\boldsymbol{bw}_1 = repeat - column(\boldsymbol{bw}, M) \in \mathcal{R}_{\geq 0}^{N \times M}, \ \boldsymbol{THP}_1 = repeat - row(\boldsymbol{THP}, N) \in \mathcal{R}_{\geq 0}^{N \times M}$$

Next, we concatenate $\boldsymbol{b}_1$, $\boldsymbol{v}_1$, $\boldsymbol{bw}_1$, $\boldsymbol{BW}$ and $\boldsymbol{THP}_1$ to obtain a 3-dimensional tensor that contains global input information; this tensor is denoted as $E \in \mathcal{R}^{N \times M \times 5}$ and is used as the input of the Transformer-Interaction Layer:

$$E = concat\left[\boldsymbol{b}_1; \boldsymbol{v}_1; \boldsymbol{bw}_1; \boldsymbol{BW}; \boldsymbol{THP}_1\right]$$

Transformer-Interaction Layer The input of the Transformer-Interaction Layer is a 3-dimensional tensor $E \in \mathcal{R}^{N \times M \times 5}$ that contains global input information. We use two 1×1 convolutions with a rectified linear unit ($ReLU$) activation function to increase the third dimension of E from 5 to D (the 1×1 convolutions do not destroy the permutation-equivariant property), which is denoted as $ReLU\left(x\right) := \max\left(0, x\right)$:

$$A = conv_2\left(conv_1\left(E\right)\right) \in \mathcal{R}^{N \times M \times D}$$

To obtain the task allocation scheme, payment scheme, and bandwidth allocation scheme, we let the edge server interact with its connected users via the Transformer to obtain $AC_j \in \mathcal{R}^{N \times D}$, $\forall j \in \mathcal{M}$:

$$AC_j = Transformer - column\left(A_{*,j}\right) \in \mathcal{R}^{N \times D}, \ \forall j \in \mathcal{M}$$
$$AC = concat\left[AC_1, AC_2, ..., AC_M\right] \in \mathcal{R}^{N \times M \times D}$$

We let the user interact with its connected edge servers via the Transformer to obtain $AC_i \in \mathcal{R}^{M\times D}$, $\forall i \in \mathcal{N}$:

$$AR_i = Transformer - row\left(A_{i,*}\right) \in \mathcal{R}^{M\times D},\ \forall i \in \mathcal{N}$$
$$AR = concat\left[AR_1, AR_2, ..., AR_N\right] \in \mathcal{R}^{N\times M\times D}$$

Finally, we concatenate the interaction result AC with AR to obtain a 3-dimensional tensor $ACR \in \mathcal{R}^{N\times M\times 2*D}$ that contains the total mutual interaction information between the users and edge servers:

$$ACR = concat\left[AC; AR\right] \in \mathcal{R}^{N\times M\times 2*D}$$

The above material describes the network architecture of a single Transformer-Interaction Layer. We can stack multiple Transformer-Interaction Layers to enable SVRANet to acquire sufficient interaction information. When stacking multiple transformer interaction layers, we utilize two 1×1 convolutions to reduce the third dimension of ACR. In the final layer, we reduce the third dimension to 3, obtaining a global feature map $F \in \mathcal{R}^{N\times M\times 3}$ as the input for the Output Layer. Otherwise, the third dimension is reduced to D and is input into the Transformer-Interaction Layer for interaction purposes.

Output Layer The input of the Output Layer is the global feature map $F \in \mathcal{R}^{N\times M\times 3}$, and the task allocation scheme, payment scheme and bandwidth allocation scheme are calculated through F. As shown in the Output Layer in Fig. 2, we divide F into 3 parts, $Ft \in \mathcal{R}^{N\times M}$, $Fp \in \mathcal{R}^{N\times M}$ and $Fd \in \mathcal{R}^{N\times M}$, which are used to calculate the task allocation scheme, payment scheme and bandwidth allocation, respectively.

We calculate the mean of $Ft \in \mathcal{R}^{N\times M}$ in a rowwise manner and then apply the $Sigmoid$ activation function to ensure that task allocation level is within $(0,1)$, $Sigmoid\left(x\right) := \frac{1}{1+e^{-x}} \in (0,1)$:

$$a_i^w = Sigmoid\left(mean\left(Ft_{i,*}, row\right)\right) \in (0,1),\ \forall i \in \mathcal{N}$$

We calculate the mean of $Fp \in \mathcal{R}^{N\times M}$ in a rowwise fashion to obtain the payment:

$$p_i^w = mean\left(Ft_{i,*}, row\right) \in (0,1),\ \forall i \in \mathcal{N}$$

We apply the $Softmax$ activation function in a rowwise manner to ensure that the bandwidth allocation level satisfies the user's requirements, with $Softmax\left(x_j\right) := \frac{e^{x_j}}{\sum_{j=1}^{M} x_j} \in (0,1)$, $\sum_{j=1}^{M} x_j = 1$:

$$f_{i,j}^w = Softmax\left(Fb_{i,j}, row\right) \in (0,1),\ \forall i \in \mathcal{N},\ j \in \mathcal{M}$$

Appendix-3

The Impact of the Budget on Utility and Social Welfare at Different User Scales.

In terms of utility acquisition, SVRANet outperforms OPSM+CPLEX and can achieve over 65% of OPT-CPLEX, as shown in Fig. 3a,3b,3c. In terms of social welfare acquisition, SVRANet performs far better than OPSM+CPLEX and can achieve over 87% of OPT-CPLEX, as shown in Fig. 3d,3e,3f.

Experimental data

As shown in Tables 1, 2, 3, 4, 5, 6 and **??**, we present the experimental data of $u_{vsp}\left(w\right)$, $rgt\left(w\right)$, $irp\left(w\right)$, $bfp\left(w\right)$, $bwp\left(w\right)$, $thpp\left(w\right)$ and $total\ payment\left(w\right)$.

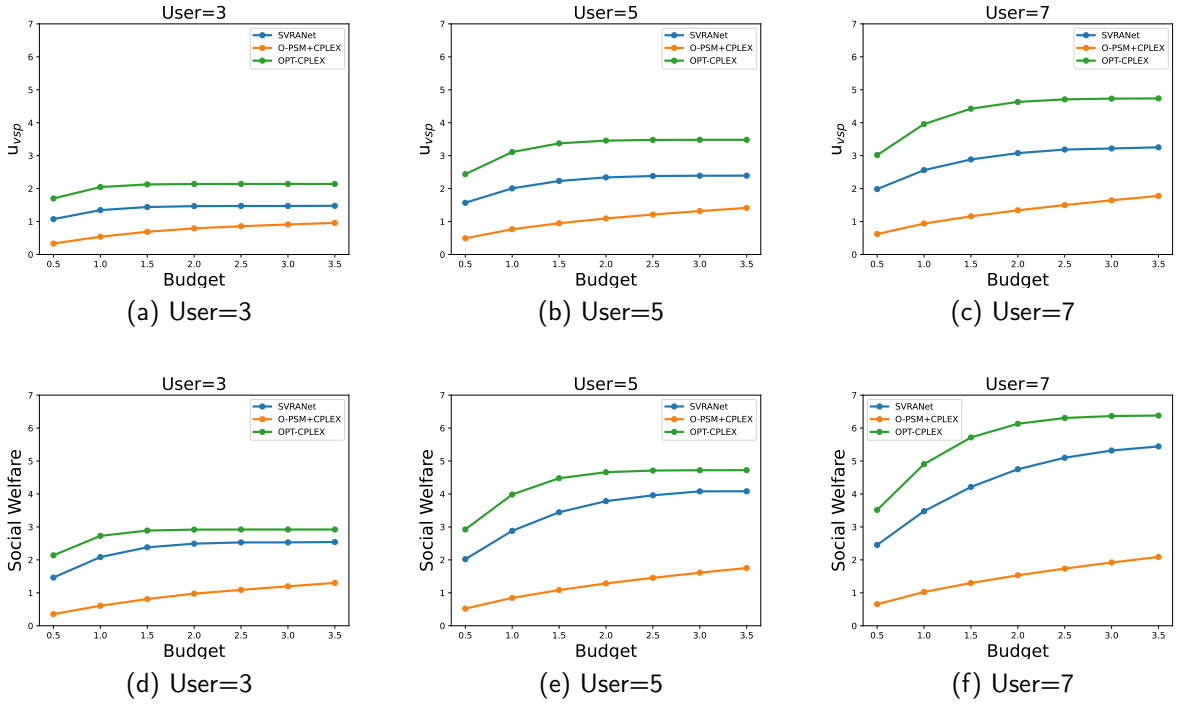

Fig. 3. The impact of the budget on Utility and Social Welfare at different user scales.

Table 1. $u_{vsp}(w)$

$user \backslash budget$	0.5	1.0	1.5	2.0	2.5	3.0	3.5
2	0.7706	0.9469	0.9836	0.9944	0.9936	0.9907	0.9929
3	1.0737	1.3474	1.4412	1.4678	1.4727	1.4732	1.4777
4	1.3417	1.6866	1.8526	1.9039	1.9257	1.9347	1.9355
5	1.5711	2.0077	2.2318	2.3407	2.3818	2.3912	2.3933
6	1.7802	2.3091	2.5581	2.7325	2.7979	2.8265	2.8261
7	1.9861	2.5604	2.8847	3.0773	3.1854	3.2185	3.2526

Table 2. $rgt(w)$

$user \backslash budget$	0.5	1.0	1.5	2.0	2.5	3.0	3.5
2	0.000332	0.000278	0.000256	0.000216	0.000214	0.000222	0.000212
3	0.000331	0.000291	0.000260	0.000226	0.000211	0.000218	0.000218
4	0.000321	0.000309	0.000265	0.000241	0.000227	0.000211	0.000224
5	0.000307	0.000305	0.000276	0.000252	0.000227	0.000226	0.000225
6	0.000302	0.000295	0.000287	0.000268	0.000244	0.000229	0.000216
7	0.000297	0.000296	0.000295	0.000271	0.000262	0.000245	0.000229

Table 3. $irp\,(w)$

$user\backslash budget$	0.5	1.0	1.5	2.0	2.5	3.0	3.5
2	0.000131	0.000191	0.000195	0.000183	0.000190	0.000179	0.000195
3	0.000120	0.000161	0.000192	0.000198	0.000182	0.000192	0.000187
4	0.000104	0.000160	0.000177	0.000187	0.000196	0.000196	0.000187
5	0.000098	0.000139	0.000167	0.000171	0.000185	0.000191	0.000186
6	0.000093	0.000129	0.000164	0.000174	0.000176	0.000183	0.000175
7	0.000083	0.000124	0.000147	0.000160	0.000179	0.000182	0.000187

Table 4. $irp\,(w)$

$user\backslash budget$	0.5	1.0	1.5	2.0	2.5	3.0	3.5
2	0.000106	0.000082	0.000065	0.0	0.0	0.0	0.0
3	0.000140	0.000097	0.000085	0.000063	0.000046	0.0	0.0
4	0.000144	0.000128	0.000100	0.000091	0.000072	0.000061	0.000035
5	0.000156	0.000130	0.000113	0.000112	0.000090	0.000075	0.000064
6	0.000163	0.000160	0.000135	0.000124	0.000113	0.000093	0.000060
7	0.000184	0.000178	0.000169	0.000140	0.000135	0.000119	0.000072

Table 5. $bwp\,(w)$

$user\backslash budget$	0.5	1.0	1.5	2.0	2.5	3.0	3.5
2	0.0000562	0.0000583	0.0000577	0.0000522	0.0000588	0.0000576	0.0000654
3	0.0000514	0.0000581	0.0000585	0.0000596	0.0000654	0.0000610	0.0000656
4	0.0000505	0.0000579	0.0000535	0.0000594	0.0000639	0.0000605	0.0000591
5	0.0000471	0.0000550	0.0000578	0.0000609	0.0000572	0.0000573	0.0000599
6	0.0000462	0.0000519	0.0000553	0.0000570	0.0000574	0.0000548	0.0000557
7	0.0000428	0.0000505	0.0000516	0.0000572	0.0000607	0.0000566	0.0000542

Table 6. $thpp\,(w)$

$user\backslash budget$	0.5	1.0	1.5	2.0	2.5	3.0	3.5
2	0.000035	0.000044	0.000047	0.000043	0.000049	0.000048	0.000046
3	0.000050	0.000060	0.000063	0.000063	0.000066	0.000072	0.000070
4	0.000056	0.000068	0.000070	0.000080	0.000082	0.000081	0.000082
5	0.000060	0.000078	0.000081	0.000082	0.000083	0.000093	0.000095
6	0.000064	0.000074	0.000086	0.000092	0.000096	0.000102	0.000108
7	0.000068	0.000083	0.000093	0.000113	0.000116	0.000119	0.000115