

LaTeX Blog

- 一、图的插入
 - 1.pdf格式的图片的插入
 - 2.eps格式的图片的插入
 - 2.1两个图并排
 - 2.2三个图并排
 - 2.3五个图并排
- 二、表的插入
- 三、算法块
 - 1.算法块的插入
- 四、参考文献系列
 - 1.正文中[引用]高亮显示+超链接
- 附录：出现的问题及解决方案
 - 1.参考文献中出现横线

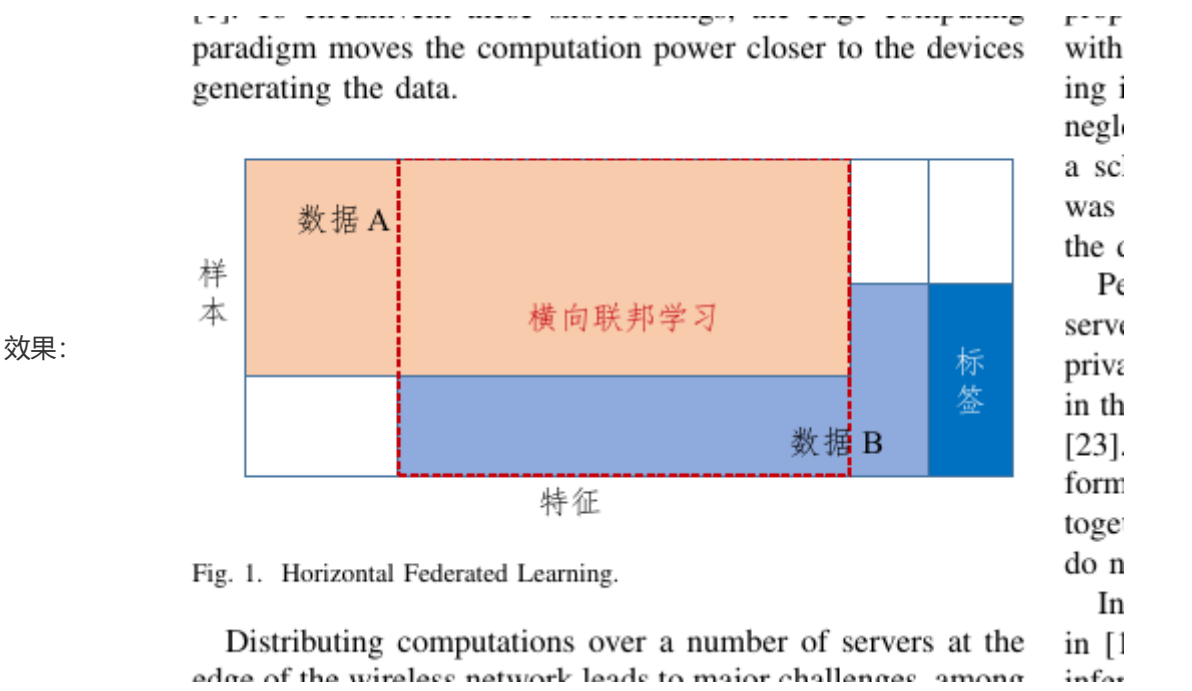
链接：

LaTeX Blog

一、图的插入

1.pdf格式的图片的插入

```
1 \begin{figure}[htb]
2   \centering
3   \includegraphics[keepaspectratio, width = 1\columnwidth]
   {Figures/Horizontal Federated Learning.pdf}
4   \caption{Horizontal Federated Learning}
5   \label{fig:Horizontal}} %用于引用的标签
6 \end{figure}
```



[h] 表示的当前位置 (here) , 也就是说图片排在你设置的当前位置, 但是如果这一页的空间不足以放下这个图片, 此时图片会转到下一页。

[t] 顶端(top)。此时系统会将图片放置在页面的顶部。

[b] 底部. (bottom) 这里是优先将图片放置在底部, 也就是页面的底部。

[p] 这个是将图片设置为浮动状态, 也就是可以根据系统排版的, 自动放置图片的位置。

2.eps格式的图片的插入

2.1两个图并排

```

1 \begin{figure}[!ht]
2   \centerline{$\begin{array}{cc}
3     \includegraphics[width=1.4in]{unsecure1.eps} &
4     \hspace{-1mm}\includegraphics[width=1.6in]{unsecure2.eps}\\
5     \hspace{-1mm}\mbox{\footnotesize a)   general illustration of $
\textbf{E}$. } &
6     \hspace{-1mm}\mbox{\footnotesize d)   optimal placement of
highlighted entries.} \\
7     \end{array}$}\caption{Illustration
of RPC.} \label{fig.uns}
9 \end{figure}

```

效果:

	x_{N_1}	...	x_{N_m}
F_{M_1}	$p_1 + q_1$		$p_1 + q_m$
	\vdots	\ddots	\vdots
F_{M_m}	$p_m + q_1$...	$p_m + q_m$

	$x_{N_1} \dots x_{N_{m-2}}$	x_1	x_m
F_1			
F_2	$(m-2) \times (m-2)$ placement		
\vdots			
F_{m-1}			
F_m			

a) general illustration of E. d) optimal placement of highlighted entries.

Fig. 1. Illustration of RPC.

2.2三个图并排

```

1 \begin{figure*}[!ht]
2   \centerline{$\begin{array}{ccc}
3     \includegraphics[width=2.0in]{secure1.eps} & &
4     \hspace{-1mm}\includegraphics[width=2.0in]{secure2.eps} & &
5     \hspace{-1mm}\includegraphics[width=2.5in]{secure3.eps}\\
6     \hspace{-1mm}\mbox{\footnotesize a)   structure of
7       $ \textbf{E} $. } & &
8     \hspace{-1mm}\mbox{\footnotesize b)   structure of
9       $ \bar{\textbf{E}} $. } & &
10    \hspace{-1mm}\mbox{\footnotesize c)   structure of
11      $ \tilde{\textbf{E}} $. } \\
12    \end{array}$}\caption{The placements of highlighted entries in
SBMM.} \label{fig.sbm}
13 \end{figure*}

```

效果：

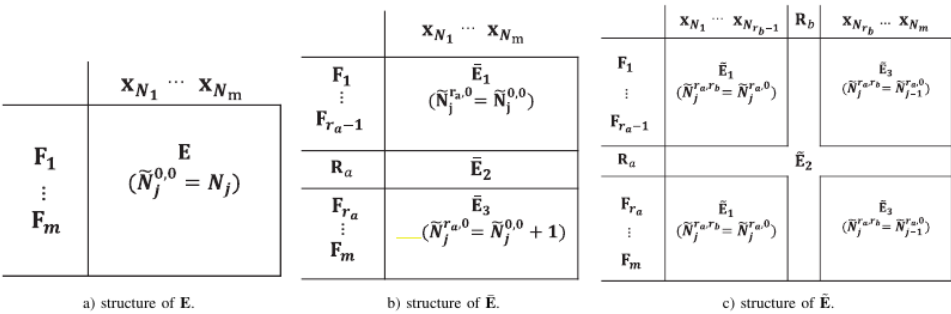


Fig. 2. The placements of highlighted entries in SBMM.

2.3五个图并排

```
1 \begin{figure*}[!ht]
2   \centerline{\$ \begin{array}{ccccc}
3     \includegraphics[width=1.3in]{ex1.eps} &
4     \hspace{-1mm} \includegraphics[width=1.3in]{ex2.eps} &
5     \hspace{-1mm} \includegraphics[width=1.3in]{ex3.eps} &
6     \hspace{-1mm} \includegraphics[width=1.5in]{ex4.eps} &
7     \hspace{-1mm} \includegraphics[width=1.5in]{ex5.eps} \\
8     \hspace{-1mm} \mbox{\footnotesize a) } \$ r_a = 0, r_b = 0 \$ .} &
9     \hspace{-1mm} \mbox{\footnotesize b) } \$ r_a = 2, r_b = 0 \$ .} &
10    \hspace{-1mm} \mbox{\footnotesize c) } \$ r_a = 3, r_b = 0 \$ .} &
11    \hspace{-1mm} \mbox{\footnotesize d) } \$ r_a = 3, r_b = 5 \$ .} &
12    \hspace{-1mm} \mbox{\footnotesize e) } \$ r_a = 3, r_b = 4 \$ .} \\
13    \end{array} \$} \caption{Examples of the placement of highlighted
14    entries.} \label{fig.ex}
15 \end{figure*}
```

效果：

Fig. 2. The placements of highlighted entries in SBMM.

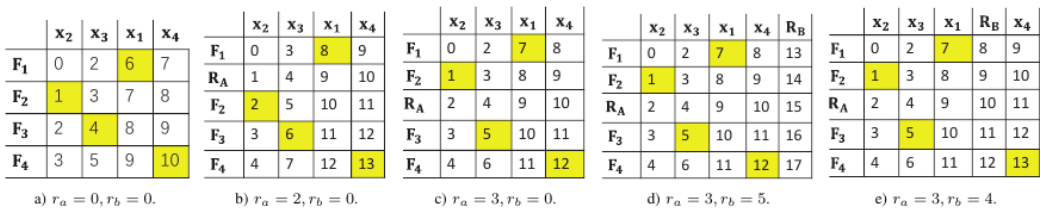


Fig. 3. Examples of the placement of highlighted entries.

二、表的插入

三、算法块

算法块需要调用包

```
1 \usepackage[ruled,linesnumbered]{algorithm2e}
```

1.算法块的插入

```
1 \begin{algorithm}\label{alg1}
2   \LinesNumbered
3   \begin{small}
4     \KwIn{$ N $, $ r_a $, $ r_b $, $ M $, $ P $}
5     \KwOut{$ Q $, $ Q_b $}
6     $ Q_1 = 0 $;\n
7     \For{$ j \ge 2 $; $ j < r_b $; $ j=j+1 $}{
8       \If {$ m+1 - \tilde{N}_{j-1}^{r_a,r_b} \le \tilde{N}_j^{r_a,r_b}-1 $}{
9         $ Q_j = Q_{j-1} + \tilde{N}_{j-1}^{r_a,r_b} $;\n
10      }
11      \Else
12      { $ Q_j = Q_{j-1} + m - \tilde{N}_j^{r_a,r_b}+2 $;\n}
13    }
14    $ Q_b = Q_{r_b-1} + \tilde{N}_{r_b-1}^{r_a,r_b} $;\n
15    $ Q_{r_b} = Q_b + m - \tilde{N}_{r_b+1}^{r_a,r_b}+2 $;\n
16    \For {$ j \ge r_b + 1 $; $ j \le m $; $ j=j+1 $}{
17      \If {$ m+1 - N_j^{r_a,r_b} \le N_{j+1}^{r_a,r_b}-1 $}{
18        $ Q_j = Q_{j-1} + N_j^{r_a,r_b} $;\n
19      }
20      \Else
21      { $ Q_j = Q_{j-1} + m - N_{j+1}^{r_a,r_b}+2 $;\n}
22    }
23
24    \Return $ Q $, $ Q_b $ ;
25    \caption{The Optimal values of $ Q $ with a Given Placement of
Highlighted Entries.} \label{optimal}
26  \end{small}
27 \end{algorithm}
```

Algorithm 1: The Optimal Values of Q with a Given Placement of Highlighted Entries.

```
Input:  $N, r_a, r_b, M, P$ 
Output:  $Q, Q_b$ 
1  $Q_1 = 0$ ;
2 for  $j \geq 2; j < r_b; j = j + 1$  do
3   if  $m + 1 - \tilde{N}_{j-1}^{r_a, r_b} \leq \tilde{N}_j^{r_a, r_b} - 1$  then
4      $Q_j = Q_{j-1} + \tilde{N}_{j-1}^{r_a, r_b}$ ;
5   end
6   else
7      $Q_j = Q_{j-1} + m - \tilde{N}_j^{r_a, r_b} + 2$ ;
8   end
9 end
10  $Q_b = Q_{r_b-1} + \tilde{N}_{r_b-1}^{r_a, r_b}$ ;
11  $Q_{r_b} = Q_b + m - \tilde{N}_{r_b+1}^{r_a, r_b} + 2$ ;
12 for  $j \geq r_b + 1; j \leq m; j = j + 1$  do
13   if  $m + 1 - \tilde{N}_j^{r_a, r_b} \leq \tilde{N}_{j+1}^{r_a, r_b} - 1$  then
14      $Q_j = Q_{j-1} + \tilde{N}_j^{r_a, r_b}$ ;
15   end
16   else
17      $Q_j = Q_{j-1} + m - \tilde{N}_{j+1}^{r_a, r_b} + 2$ ;
18   end
19 end
20 return  $Q, Q_b$  ;
```

效果：

customer has its preferences, which are encoded by an attribute vector \mathbf{x} . Based on a customer's preferences, the application

comple
latenc
scenar
the la
[16],
This
decod
of a l
rely c
privac
Furthe
the va
linear
increa
Not
upper
vector
vector
finite
We u
{1, 2,
than c
than
notati
3 \mapsto 4

四、参考文献系列

1.正文中[引用]高亮显示+超链接

```
1 | \usepackage[backref]{hyperref} %用于引用突出，会报错，但能编译
```

效果如图

B. Coded Computing

There are two main challenges in distributed computing: How to reduce communication overhead and ensure safety. In order to calculate the final result, a large number of intermediate results need to be exchanged between computing nodes through the network. This not only increases communication overhead, but also limits the performance of distributed computing applications. For example, for the Hadoop cluster on Facebook, the average time spent in the data transfer phase accounts for 33% of the entire job execution time [4]. When running the Terasort application on a heterogeneous Amazon EC2 cluster, data shuffling will take 65% of the time. Approximately 70% of the total execution time is spent on the Self-Join application [5]. In the actual training of convolutional neural networks, communication restrictions are even more embarrassing. As we all know, millions of model parameters need to be updated in ResNet-50 [6] and AlexNet [7].

- [16] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1920–1933, 2020. II-A
- [17] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, "DRACO: Byzantine-resilient distributed training via redundant gradients," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 903–912. [Online]. Available: <https://proceedings.mlr.press/v80/chen18l.html> II-A III
- [18] W.-T. Chang and R. Tandon, "On the capacity of secure distributed matrix multiplication," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6. II-B II-B II-B
- [19] R. Bitar, P. Parag, and S. El Rouayheb, "Minimizing latency for secure distributed computing," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2900–2904. II-B II-B
- [20] R. Bitar, P. Parag, and S. El Rouayheb, "Minimizing latency for secure coded computing using secret sharing via staircase codes," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4609–4619, 2020. II-B
- [21] J. Kakar, S. Ebadifar, and A. Sezgin, "Rate efficiency and straggler

附录：出现的问题及解决方案

1.参考文献中出现横线

- trix multiplication," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.
- [19] R. Bitar, P. Parag, and S. El Rouayheb, "Minimizing latency for secure distributed computing," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2900–2904.
 - [20] —, "Minimizing latency for secure coded computing using secret sharing via staircase codes," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4609–4619, 2020.
 - [21] J. Kakar, S. Ebadifar, and A. Sezgin, "Rate efficiency and straggler

原因：这是因为相邻两篇文献的作者相同，在IEEE模板下会出现横线。

找到TexLive安装目录下的“**IEEEtran.bst**”文件

```
1 | 路径：D:\Program Files\TexLive\2021\texmf-dist\bibtex\bst\IEEEtran.bst
```

找到如下代码(大约在128行)，将代码中的#1变为#0

```
1 | FUNCTION {default.is.dash.repeated.names} { #1 }
```

重新编译，解决问题

- (GLOBECOM), 2018, pp. 1–6.
- [19] R. Bitar, P. Parag, and S. El Rouayheb, "Minimizing latency for secure distributed computing," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2900–2904.
 - [20] R. Bitar, P. Parag, and S. El Rouayheb, "Minimizing latency for secure coded computing using secret sharing via staircase codes," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4609–4619, 2020.
 - [21] J. Kakar, S. Ebadifar, and A. Sezgin, "Rate-efficiency and straggler-

