

文献阅读分享

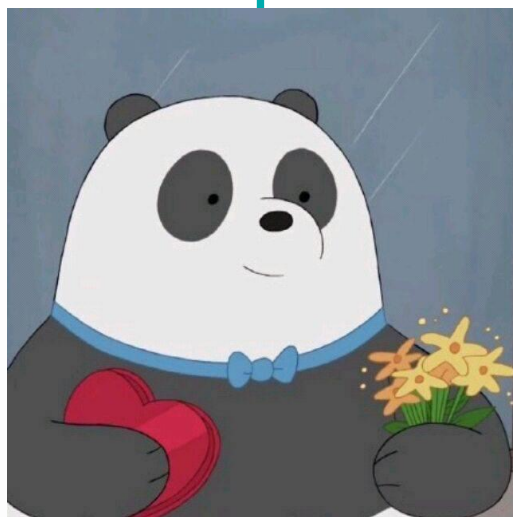
王沛然

目录

- 文献阅读的重要性
- 文献查找
- 文献阅读的方法
- 文献笔记



竞赛
科研组



■ 姓名： 王沛然

■ 年级： 2018级

■ 专业： 网络空间安全学院 卓越人才班

■ 辅导方向：

科研入门；计软网科研方向选择；保研留学

■ 问题示例：

1.计软网研究方向如何选择；2.如何进课题组

■ 个人经历：

国家奖学金和综合一等奖学金；一篇CCF-A期刊在投，两篇CCF-C会议，一篇EI会议发表，曾获会议best paper award；微软亚洲研究院系统研究组研究实习；约翰·霍普金斯大学研究实习；雅思自学7.5，GRE自学326；国家级大创、省级大创、互联网+省银、校金、校银参与者

文献阅读的重要性

为什么文献阅读这么重要？我为什么不能直接A上去？ 😞

为什么要读论文？

- 读论文为什么重要：
 - 提升自己知识深度，跟上前沿研究的深度
 - 了解最新的研究热点
 - 了解学术圈生态
 - 有哪些比较出名的组？
 - 有哪些常在这个领域发论文的组？
 - 跟踪本领域牛组对这个问题的研究路线演化
 - 提高阅读和写作水平

文献查找

文献那么多？我应该怎么查找呢？找到文献后如何区分哪些需要精读，哪些需要粗读呢？🤔

如何查找文献？

- 文献搜索平台：
 - 中文：
 - 知网、百度学术
 - 英文：
 - Google Scholar(放镜像网页) , Research Gate
 - 本领域的著名出版商, 比如计算机领域的ACM, 电子电气计算机领域的IEEE
- 查找方法：
 - 相关关键词搜索
 - 在已有的论文的引用文献和引用改论文的文献找
 - 综述

利用引用文献查找

Google 学术搜索

文章 找到约 360 条结果 (用时0.02秒)

时间不限
2022以来
2021以来
2018以来
自定义范围...

按相关性排序
按日期排序

不限语言
中文网页
简体中文网页

创建快讯

Megatron-Lm: Training multi-billion parameter language models using model parallelism
在引用文章中搜索

Huggingface's transformers: State-of-the-art natural language processing [PDF] arxiv.org
CODE
T Wolf, L Debut, V Sanh, J Chaumond... - arXiv preprint arXiv ..., 2019 - arxiv.org
Recent progress in natural language processing has been driven by advances in both model architecture and model pretraining. Transformer architectures have facilitated ...
☆ 保存 99 引用 被引用次数: 1003 相关文章 所有 4 个版本

[HTML] Transformers: State-of-the-art natural language processing [HTML] aclanthology.org
T Wolf, L Debut, V Sanh, J Chaumond... - Proceedings of the ..., 2020 - aclanthology.org
Recent progress in natural language processing has been driven by advances in both model architecture and model pretraining. Transformer architectures have facilitated ...
☆ 保存 99 引用 被引用次数: 768 相关文章 所有 6 个版本

On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? [PDF] acm.org
E M Bender, T Gebru, A McMillan-Major... - Proceedings of the 2021 ..., 2021 - dl.acm.org
The past 3 years of work in NLP have been characterized by the development and deployment of ever larger language models, especially for English: BERT, its variants, GPT ...
☆ 保存 99 引用 被引用次数: 459 相关文章 所有 8 个版本

Pre-trained models for natural language processing: A survey [PDF] arxiv.org
X Qiu, T Sun, Y Xu, Y Shao, N Dai, X Huang - Science China ..., 2020 - Springer
Recently, the emergence of pre-trained models (PTMs) has brought natural language processing (NLP) to a new era. In this survey, we provide a comprehensive review of PTMs ...
☆ 保存 99 引用 被引用次数: 333 相关文章 所有 9 个版本

Recipes for building an open-domain chatbot [PDF] arxiv.org
S Roller, E Dinan, N Goyal, D Ju, M Williamson... - arXiv preprint arXiv ..., 2020 - arxiv.org
Building open-domain chatbots is a challenging area for machine learning research. While prior work has shown that scaling neural models in the number of parameters and the size of ...
☆ 保存 99 引用 被引用次数: 260 相关文章 所有 6 个版本

DeBERTa: Decoding-enhanced bert with disentangled attention [PDF] arxiv.org
P He, X Liu, J Gao, W Chen - arXiv preprint arXiv:2006.03654, 2020 - arxiv.org

利用引用文献查找

Figure 2. Transformer Architecture. Purple blocks correspond to fully connected layers. Each blue block represents a single transformer layer that is replicated N times.

and compute efficiency. The original transformer formulation was designed as a machine translation architecture that transforms an input sequence into another output sequence using two parts, an *Encoder* and *Decoder*. However, recent work leveraging transformers for language modeling such as BERT (Devlin et al., 2018) and GPT-2 (Radford et al., 2019) use only the *Encoder* or *Decoder* depending on their needs. This work explores both a decoder architecture, GPT-2, and an encoder architecture, BERT.

Figure 2 shows a schematic diagram of the model we used. We refer the reader to prior work for a detailed description of the model architecture (Vaswani et al., 2017; Devlin et al., 2018; Radford et al., 2019). It is worthwhile to mention that both GPT-2 and BERT use GeLU (Hendrycks & Gimpel, 2016) nonlinearities and layer normalization (Ba et al., 2016) to the input of the multi-head attention and feed forward layers, whereas the original transformer (Vaswani et al., 2017) uses ReLU nonlinearities and applies layer normalization to outputs.

2.3. Data and Model Parallelism in Deep Learning

There are two central paradigms for scaling out deep neural network training to numerous hardware accelerators: data parallelism (Valiant, 1990) where a training minibatch is split across multiple workers, and model parallelism in which the memory usage and computation of a model is distributed across multiple workers. By increasing the minibatch size proportionally to the number of available workers (i.e. *weak scaling*), one observes near linear scaling in training data throughput. However, large batch training introduces complications into the optimization process that can result in reduced accuracy or longer time to convergence, offsetting the benefit of increased training throughput (Keskar et al., 2017). Further research (Goyal et al., 2017; You et al., 2017; 2019) has developed techniques to miti-

model (Lan et al., 2019), but this limits the overall capacity of the model. Our approach is to utilize model parallelism to split the model across multiple accelerators. This not only alleviates the memory pressure, but also increases the amount of parallelism independently of the microbatch size.

Within model parallelism, there are two further paradigms: layer-wise pipeline parallelism, and more general distributed tensor computation. In pipeline model parallelism, groups of operations are performed on one device before the outputs are passed to the next device in the pipeline where a different group of operations are performed. Some approaches (Harlap et al., 2018; Chen et al., 2018) use a parameter server (Li et al., 2014) in conjunction with pipeline parallelism. However these suffer from inconsistency issues. The GPipe framework for TensorFlow (Huang et al., 2018) overcomes this inconsistency issue by using synchronous gradient descent. This approach requires additional logic to handle the efficient pipelining of these communication and computation operations, and suffers from pipeline bubbles that reduce efficiency, or changes to the optimizer itself which impact accuracy.

Distributed tensor computation is an orthogonal and more general approach that partitions a tensor operation across multiple devices to accelerate computation or increase model size. FlexFlow (Jia et al., 2018), a deep learning framework orchestrating such parallel computation, provides a method to pick the best parallelization strategy. Recently, Mesh-TensorFlow (Shazeer et al., 2018) introduced a language for specifying a general class of distributed tensor computations in TensorFlow (Abadi et al., 2015). The parallel dimensions are specified in the language by the end user and the resulting graph is compiled with proper collective primitives. We utilize similar insights to those leveraged in Mesh-TensorFlow and exploit parallelism in computing the transformer's attention heads to parallelize our transformer model. However, rather than implementing a framework and compiler for model parallelism, we make only a few targeted modifications to existing PyTorch transformer implementations. Our approach is simple, does not

什么是精读和粗读


什么是粗读？

- 目的：了解论文解决的问题，研究的方法，**不细究方法的细节问题**
- 适用范围：和我们的研究方向**间接相关**或者**有那么一点关系**但**关系不大**的论文

什么是精读

- 目的：**细致的了解论文**的**整个脉络**，研究思路及方法，乃至使用的数据来源等
- 适用范围：与我们的研究方向**十分相关**的论文，乃至**我们的研究基于的论文**

文献阅读的方法

现在我要读一篇文献了，这么多公式我要全部弄懂吗？论文需要我全部看完吗？

如何读一篇文献

Summing up inequalities (29) and (30), we have:

$$\begin{aligned} & \langle \mathbf{w}^* - \mathbf{w}^{t-1}, \nabla F(\mathbf{w}^{t-1}) - \nabla F(\mathbf{w}^*) \rangle \\ & \leq -\frac{\mu}{2} \|\mathbf{w}^{t-1} - \mathbf{w}^*\|^2. \end{aligned} \quad (31)$$

Substituting inequalities (28) and (31) into (27), we have:

$$\begin{aligned} & \|\mathbf{w}^{t-1} - \mathbf{w}^* - \alpha \nabla F(\mathbf{w}^{t-1})\|^2 \\ & \leq (1 + \alpha^2 L^2 - \alpha \mu) \|\mathbf{w}^{t-1} - \mathbf{w}^*\|^2. \end{aligned} \quad (32)$$

By choosing $\alpha = \mu/(2L^2)$, we have:

$$\begin{aligned} & \|\mathbf{w}^{t-1} - \mathbf{w}^* - \alpha \nabla F(\mathbf{w}^{t-1})\|^2 \\ & \leq (1 - \mu^2/(4L^2)) \|\mathbf{w}^{t-1} - \mathbf{w}^*\|^2, \end{aligned} \quad (33)$$

which concludes the proof. \blacksquare

Lemma 3. Suppose Assumption 2 holds. For any $\delta \in (0, 1)$ and any $\mathbf{w} \in \Theta$, we let $\Delta_1 = \frac{\sqrt{2\sigma_1} \sqrt{(d \log 6 + \log(3/\delta))/|D_0|}}{\sqrt{2\sigma_1} \sqrt{(d \log 6 + \log(3/\delta))/|D_0|}}$ and $\Delta_3 = \sqrt{2\sigma_2} \sqrt{(d \log 6 + \log(3/\delta))/|D_0|}$. If $\Delta_1 \leq \sigma_1^2/\gamma_1$ and $\Delta_3 \leq \sigma_2^2/\gamma_2$, then we have:

$$\begin{aligned} & \Pr \left\{ \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}^*) - \nabla F(\mathbf{w}^*) \right\| \geq 2\Delta_1 \right\} \leq \frac{\delta}{3}, \\ & \Pr \left\{ \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla h(X_i, \mathbf{w}) - \mathbb{E}[h(X, \mathbf{w})] \right\| \right. \\ & \quad \left. \geq 2\Delta_3 \|\mathbf{w} - \mathbf{w}^*\| \right\} \leq \frac{\delta}{3}. \end{aligned}$$

Proof: We prove the first inequality of Lemma 3. The proof of the second inequality is similar, and we omit it for brevity. Let $\mathbf{V} = \{v_1, \dots, v_{N_2}\}$ be an $\frac{1}{2}$ -cover of the unit sphere \mathbf{B} . It is shown in [12], [39] that we have $\log N_2 \leq d \log 6$ and the following:

$$\begin{aligned} & \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}^*) - \nabla F(\mathbf{w}^*) \right\| \leq \\ & 2 \sup_{v \in \mathbf{V}} \left\{ \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}^*) - \nabla F(\mathbf{w}^*), v \right\| \right\}. \end{aligned} \quad (34)$$

According to the concentration inequalities for sub-exponential random variables [40], when Assumption 2 and condition $\Delta_1 \leq \sigma_1^2/\gamma_1$ hold, we have:

$$\begin{aligned} & \Pr \left\{ \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}^*) - \nabla F(\mathbf{w}^*), v \right\| \geq \Delta_1 \right\} \\ & \leq \exp(-|D_0| \Delta_1^2/(2\sigma_1^2)). \end{aligned} \quad (35)$$

Taking the union bound over all vectors in \mathbf{V} and combining it with inequality (34), we have:

$$\Pr \left\{ \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}^*) - \nabla F(\mathbf{w}^*) \right\| \geq 2\Delta_1 \right\}$$

$$\leq \exp(-|D_0| \Delta_1^2/(2\sigma_1^2) + d \log 6). \quad (36)$$

We conclude the proof by letting $\Delta_1 = \frac{\sqrt{2\sigma_1} \sqrt{(d \log 6 + \log(3/\delta))/|D_0|}}{\sqrt{2\sigma_1} \sqrt{(d \log 6 + \log(3/\delta))/|D_0|}}$ in (36). \blacksquare

Lemma 4. Suppose Assumptions 1-3 hold and $\Theta \subset \{\mathbf{w} : \|\mathbf{w} - \mathbf{w}^*\| \leq r\sqrt{d}\}$ holds for some positive parameter r . Then, for any $\delta \in (0, 1)$, if $\Delta_1 \leq \sigma_1^2/\gamma_1$ and $\Delta_2 \leq \sigma_2^2/\gamma_2$, we have the following for any $\mathbf{w} \in \Theta$:

$$\begin{aligned} & \Pr \{\|g_0 - \nabla F(\mathbf{w})\| \leq 8\Delta_2 \|\mathbf{w} - \mathbf{w}^*\| + 4\Delta_1\} \geq 1 - \delta, \\ & \text{where } \Delta_2 = \sigma_2 \sqrt{\frac{2}{|D_0|}} \sqrt{K_1 + K_2}, K_1 = d \log \frac{18L_2}{\sigma_2}, K_2 = \\ & \frac{1}{2} d \log \frac{|D_0|}{d} + \log \left(\frac{6\sigma_2^2 r \sqrt{|D_0|}}{\gamma_2 \sigma_1 \delta} \right), L_2 = \max\{L, L_1\}, \text{ and } |D_0| \\ & \text{is the size of the root dataset.} \end{aligned}$$

Proof: Our proof is mainly based on the ϵ -net argument [39] and [12]. We let $\tau = \frac{2\sigma_2}{\sigma_1} \sqrt{\frac{d}{|D_0|}}$ and ℓ^* be an integer that satisfies $\ell^* = \lceil r\sqrt{d}/\tau \rceil$. For any integer $1 \leq \ell \leq \ell^*$, we define $\Theta_\ell = \{\mathbf{w} : \|\mathbf{w} - \mathbf{w}^*\| \leq \tau\ell\}$. Given an integer ℓ , we let $\mathbf{w}_1, \dots, \mathbf{w}_{N_\ell}$ be an ϵ_ℓ -cover of Θ_ℓ , where $\epsilon_\ell = \frac{2\sigma_2}{\sigma_1} \sqrt{\frac{d}{|D_0|}}$ and $L_2 = \max\{L, L_1\}$. From [39], we know that $\log N_\ell \leq d \log \left(\frac{3\tau\ell}{\epsilon_\ell} \right)$. For any $\mathbf{w} \in \Theta_\ell$, there exists a j_ℓ ($1 \leq j_\ell \leq N_\ell$) such that:

$$\|\mathbf{w} - \mathbf{w}_{j_\ell}\| \leq \epsilon_\ell. \quad (37)$$

According to the triangle inequality, we have:

$$\begin{aligned} & \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}) - \nabla F(\mathbf{w}) \right\| \leq \|\nabla F(\mathbf{w}) - \nabla F(\mathbf{w}_{j_\ell})\| \\ & + \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} (\nabla f(X_i, \mathbf{w}) - \nabla f(X_i, \mathbf{w}_{j_\ell})) \right\| \\ & + \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}_{j_\ell}) - \nabla F(\mathbf{w}_{j_\ell}) \right\|. \end{aligned} \quad (38)$$

According to Assumption 1 and inequality (37), we have:

$$\|\nabla F(\mathbf{w}) - \nabla F(\mathbf{w}_{j_\ell})\| \leq L \|\mathbf{w} - \mathbf{w}_{j_\ell}\| \leq L\epsilon_\ell \quad (39)$$

Next, we define an event \mathcal{E}_1 as follows:

$$\mathcal{E}_1 = \left\{ \sup_{\mathbf{w}, \hat{\mathbf{w}} \in \Theta, \mathbf{w} \neq \hat{\mathbf{w}}} \frac{\|\nabla f(X, \mathbf{w}) - \nabla f(X, \hat{\mathbf{w}})\|}{\|\mathbf{w} - \hat{\mathbf{w}}\|} \leq L_1 \right\}.$$

According to Assumption 1, we have $\Pr\{\mathcal{E}_1\} \geq 1 - \frac{\delta}{2}$. Moreover, we have the following:

$$\begin{aligned} & \sup_{\mathbf{w} \in \Theta} \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} (\nabla f(X_i, \mathbf{w}) - \nabla f(X_i, \mathbf{w}_{j_\ell})) \right\| \\ & \leq L_1 \|\mathbf{w} - \mathbf{w}_{j_\ell}\| \leq L_1 \epsilon_\ell. \end{aligned} \quad (40)$$

According to the triangle inequality, we have:

$$\begin{aligned} & \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}_{j_\ell}) - \nabla F(\mathbf{w}_{j_\ell}) \right\| \\ & \leq \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}^*) - \nabla F(\mathbf{w}^*) \right\| \\ & + \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} (\nabla f(X_i, \mathbf{w}_{j_\ell}) - \nabla f(X_i, \mathbf{w}^*)) \right. \\ & \quad \left. - (\nabla F(\mathbf{w}_{j_\ell}) - \nabla F(\mathbf{w}^*)) \right\| \\ & \stackrel{(a)}{\leq} \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}^*) - \nabla F(\mathbf{w}^*) \right\| \\ & + \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} h(X_i, \mathbf{w}_{j_\ell}) - \mathbb{E}[h(X, \mathbf{w}_{j_\ell})] \right\|, \end{aligned} \quad (41)$$

where (a) is due to $\mathbb{E}[h(X, \mathbf{w})] = \nabla F(\mathbf{w}) - \nabla F(\mathbf{w}^*)$.

We also define events \mathcal{E}_2 and $\mathcal{E}_3(\ell)$ as:

$$\begin{aligned} \mathcal{E}_2 &= \left\{ \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}^*) - \nabla F(\mathbf{w}^*) \right\| \leq 2\Delta_1 \right\}, \\ \mathcal{E}_3(\ell) &= \left\{ \sup_{1 \leq j \leq N_\ell} \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} h(X_i, \mathbf{w}_j) - \mathbb{E}[h(X, \mathbf{w}_j)] \right\| \right. \\ & \quad \left. \leq 2\Delta_2 \tau \ell \right\}. \end{aligned}$$

According to Lemma 3 and [12], $\Delta_1 \leq \sigma_1^2/\gamma_1$, and $\Delta_2 \leq \sigma_2^2/\gamma_2$, we have $\Pr\{\mathcal{E}_2\} \geq 1 - \frac{\delta}{2}$ and $\Pr\{\mathcal{E}_3(\ell)\} \geq 1 - \frac{\delta}{3\ell^2}$. Therefore, on event $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3(\ell)$, we have:

$$\begin{aligned} & \sup_{\mathbf{w} \in \Theta_\ell} \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}) - \nabla F(\mathbf{w}) \right\| \\ & \leq L\epsilon_\ell + L_1\epsilon_\ell + 2\Delta_1 + 2\Delta_2\tau\ell, \end{aligned} \quad (42)$$

$$\stackrel{(a)}{\leq} 2L_2\epsilon_\ell + 2\Delta_1 + 2\Delta_2\tau\ell \stackrel{(b)}{\leq} 4\Delta_2\tau\ell + 2\Delta_1, \quad (43)$$

where (a) holds because $(L + L_1) \leq 2L_2$ and (b) is due to $\Delta_2 \geq \sigma_2 \sqrt{d/|D_0|}$.

Thus, according to the union bound, we have probability at least $1 - \delta$ that event $\mathcal{E}_1 \cap \mathcal{E}_2 \cap (\cap_{\ell=1}^{\ell^*} \mathcal{E}_3(\ell))$ holds. On event $\mathcal{E}_1 \cap \mathcal{E}_2 \cap (\cap_{\ell=1}^{\ell^*} \mathcal{E}_3(\ell))$, for any $\mathbf{w} \in \Theta_{\ell^*}$, there exists an $1 \leq \ell \leq \ell^*$ such that $(\ell - 1)\tau < \|\mathbf{w} - \mathbf{w}^*\| \leq \ell\tau$ holds. If $\ell = 1$, then we have:

$$\left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}) - \nabla F(\mathbf{w}) \right\|$$

$$\leq 4\Delta_2\tau + 2\Delta_1 \stackrel{(a)}{\leq} 4\Delta_1, \quad (44)$$

where (a) holds because $\Delta_2 \leq \sigma_2^2/\gamma_2$ and $\Delta_1 \geq \sigma_1 \sqrt{d/|D_0|}$. If $\ell \geq 2$, then we have $2(\ell - 1) \geq \ell$ and the following:

$$\begin{aligned} & \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}) - \nabla F(\mathbf{w}) \right\| \\ & \leq 8\Delta_2 \|\mathbf{w} - \mathbf{w}^*\| + 2\Delta_1. \end{aligned} \quad (45)$$

Combining inequalities (44) and (45), we have:

$$\begin{aligned} & \sup_{\mathbf{w} \in \Theta_{\ell^*}} \left\| \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w}) - \nabla F(\mathbf{w}) \right\| \\ & \leq 8\Delta_2 \|\mathbf{w} - \mathbf{w}^*\| + 4\Delta_1. \end{aligned} \quad (46)$$

We conclude the proof since $\Theta \subset \Theta_{\ell^*}$ and $g_0 = \frac{1}{|D_0|} \sum_{X_i \in D_0} \nabla f(X_i, \mathbf{w})$. \blacksquare

Proof of Theorem 1: With the lemmas above, we can prove Theorem 1 next. We have the following equations for the t th global iteration:

$$\begin{aligned} & \|\mathbf{w}^t - \mathbf{w}^*\| \\ &= \|\mathbf{w}^{t-1} - \alpha \mathbf{g}^{t-1} - \mathbf{w}^*\| \\ &= \|\mathbf{w}^{t-1} - \alpha \nabla F(\mathbf{w}^{t-1}) - \mathbf{w}^* + \alpha \nabla F(\mathbf{w}^{t-1}) - \alpha \mathbf{g}^{t-1}\| \\ &\leq \|\mathbf{w}^{t-1} - \alpha \nabla F(\mathbf{w}^{t-1}) - \mathbf{w}^*\| + \alpha \|\mathbf{g}^{t-1} - \nabla F(\mathbf{w}^{t-1})\| \\ &\stackrel{(a)}{\leq} \|\mathbf{w}^{t-1} - \alpha \nabla F(\mathbf{w}^{t-1}) - \mathbf{w}^*\| + 3\alpha \|\mathbf{g}_0^{t-1} - \nabla F(\mathbf{w}^{t-1})\| \\ &\quad + 2\alpha \|\nabla F(\mathbf{w}^{t-1})\| \\ &\stackrel{(b)}{\leq} \underbrace{\|\mathbf{w}^{t-1} - \alpha \nabla F(\mathbf{w}^{t-1}) - \mathbf{w}^*\|}_{A_1} + 3\alpha \underbrace{\|\mathbf{g}_0^{t-1} - \nabla F(\mathbf{w}^{t-1})\|}_{A_2} \\ &\quad + 2\alpha \underbrace{\|\nabla F(\mathbf{w}^{t-1}) - \nabla F(\mathbf{w}^*)\|}_{A_3} \\ &\stackrel{(c)}{\leq} \sqrt{1 - \mu^2/(4L^2)} \|\mathbf{w}^{t-1} - \mathbf{w}^*\| + 2\alpha L \|\mathbf{w}^{t-1} - \mathbf{w}^*\| \\ &\quad + 3\alpha (8\Delta_2 \|\mathbf{w}^{t-1} - \mathbf{w}^*\| + 4\Delta_1) \\ &= \left(\sqrt{1 - \mu^2/(4L^2)} + 2\alpha L \right) \|\mathbf{w}^{t-1} - \mathbf{w}^*\| \\ &\quad + 12\alpha \Delta_1, \end{aligned} \quad (47)$$

where (a) is obtained based on Lemma 1; (b) is due to $\nabla F(\mathbf{w}^*) = 0$; and (c) is obtained by plugging Lemma 2, Lemma 4, and Assumption 1 into A_1 , A_2 , and A_3 , respectively. By recursively applying the inequality for each global iteration, we have:

$$\|\mathbf{w}^t - \mathbf{w}^*\| \leq (1 - \rho)^t \|\mathbf{w}^0 - \mathbf{w}^*\| + 12\alpha \Delta_1 / \rho, \quad (48)$$

where $\rho = 1 - \left(\sqrt{1 - \mu^2/(4L^2)} + 2\alpha L \right)$. Thus, we conclude the proof.

如何读一篇论文

- 一些误区：
- 读论文要从头到尾读
- 读论文要全部看懂才算读懂
- 论文的公式要全部推一遍

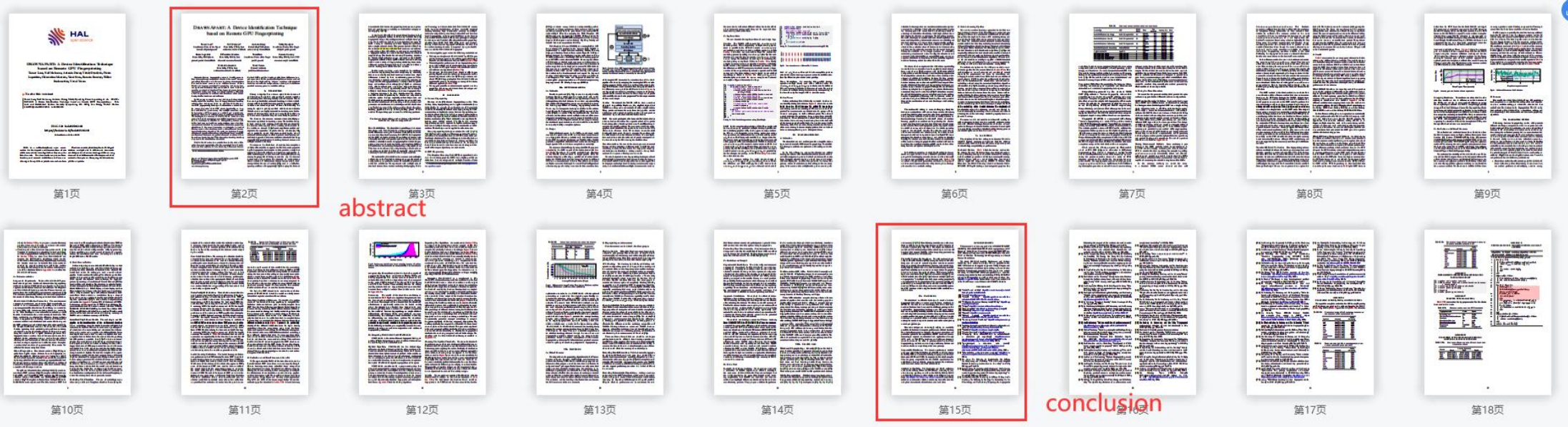
论文大致结构

- 论文结构一般按照IMRD结构：
- *abstract: 摘要，论文最开头，概括性地讲述整个论文的工作
- Introduction: 介绍论文的背景
- *Related Work: 介绍与论文相关的之前的工作
- Methods: 阐述论文所采用的主要方法
- Results: 介绍实验部分的设置，和实验结果
- Discussion: 根据实验结果总结分析整个工作的作用和意义

如何读一篇论文

- 先读abstract
- 再读contributions (introduction)
- 再读conclusion
- 读主体内容 (精读)
 - 看框架图 (Methods)
 - 看实验图
 - 读正文

如何读一篇论文



Abstract (摘要) - 粗读、精读

Abstract—Browser fingerprinting aims to identify users or their devices, through scripts that execute in the users' browser and collect information on software or hardware characteristics. It is used to track users or as an additional means of identification to improve security. Fingerprinting techniques have one significant limitation: they are unable to track individual users for an extended duration. This happens because browser fingerprints evolve over time, and these evolutions ultimately cause a fingerprint to be confused with those from other devices sharing similar hardware and software.

In this paper, we report on a new technique that can significantly extend the tracking time of fingerprint-based tracking methods. Our technique, which we call DRAWNPART, is a new *GPU fingerprinting* technique that identifies a device from the unique properties of its GPU stack. Specifically, we show that variations in speed among the multiple execution units that comprise a GPU can serve as a reliable and robust device signature, which can be collected using unprivileged JavaScript. We investigate the accuracy of DRAWNPART under two scenarios. In the first scenario, our controlled experiments confirm that the technique is effective in distinguishing devices with similar hardware and software configurations, even when they are considered identical by current state-of-the-art fingerprinting algorithms. In the second scenario, we integrate a *one-shot learning* version of our technique into a state-of-the-art browser fingerprint tracking algorithm. We verify our technique through a large-scale experiment involving data collected from over 2,500 crowd-sourced devices over a period of several months and show it provides a boost of up to 67% to the median tracking duration, compared to the state-of-the-art method.

DRAWNPART makes two contributions to the state of the art in browser fingerprinting. On the conceptual front, it is the first work that explores the manufacturing differences between

identical GPUs and the first to exploit these differences in a privacy context. On the practical front, it demonstrates a robust technique for distinguishing between machines with identical hardware and software configurations, a technique that delivers practical accuracy gains in a realistic setting.

I. INTRODUCTION

Privacy is dignity. It is a human right. In the domain of web browsing, the right to privacy should prevent websites from tracking user browsing activity without consent. This is the case in particular for cross-site tracking, in which website owners collude to build browsing profiles spanning multiple websites over extended periods of time. Unfortunately for users, the right to privacy conflicts with business interests. Website owners are highly interested in tracking users for the purpose of showing them ads they are more likely to click on, or to recommend products they are more likely to purchase.

We focus on the common scenario where identifying a browser is equivalent to tracking a user. The traditional way to track users is with cookies, small files that are stored by the browser at the request of the website, and forwarded to the website on demand [50]. Recent regulations restrict and supervise the acquisition of private data by websites [4, 31], and in particular require that users consent to the use of cookies. Furthermore, in an effort to protect users' privacy and curb tracking, modern browsers restrict cookie-based tracking, especially *third-party trackers* that attempt to track users across multiple unrelated websites.

To overcome the limitations of cookies, less scrupulous websites often resort to an approach called *browser fingerprinting*. To fingerprint a browser, the website provides a script that

- 读论文先看Abstract部分
- Abstract是整个论文的核心部分
- 作者在abstract中用比较精炼的语言概括整篇文章的背景、工作、实验和结论
- 如果只是粗读，读完abstract就可以了

Conclusion (结论) - 精读

Abstract—Browser fingerprinting aims to identify users or their devices, through scripts that execute in the users' browser and collect information on software or hardware characteristics. It is used to track users or as an additional means of identification to improve security. Fingerprinting techniques have one significant limitation: they are unable to track individual users for an extended duration. This happens because browser fingerprints evolve over time, and these evolutions ultimately cause a fingerprint to be confused with those from other devices sharing similar hardware and software.

In this paper, we report on a new technique that can significantly extend the tracking time of fingerprint-based tracking methods. Our technique, which we call DRAWNPART, is a new *GPU fingerprinting* technique that identifies a device from the unique properties of its GPU stack. Specifically, we show that variations in speed among the multiple execution units that comprise a GPU can serve as a reliable and robust device signature, which can be collected using unprivileged JavaScript. We investigate the accuracy of DRAWNPART under two scenarios. In the first scenario, our controlled experiments confirm that the technique is effective in distinguishing devices with similar hardware and software configurations, even when they are considered identical by current state-of-the-art fingerprinting algorithms. In the second scenario, we integrate a *one-shot learning* version of our technique into a state-of-the-art browser fingerprint tracking algorithm. We verify our technique through a large-scale experiment involving data collected from over 2,500 crowd-sourced devices over a period of several months and show it provides a boost of up to 67% to the median tracking duration, compared to the state-of-the-art method.

DRAWNPART makes two contributions to the state of the art in browser fingerprinting. On the conceptual front, it is the first work that explores the manufacturing differences between

identical GPUs and the first to exploit these differences in a privacy context. On the practical front, it demonstrates a robust technique for distinguishing between machines with identical hardware and software configurations, a technique that delivers practical accuracy gains in a realistic setting.

I. INTRODUCTION

Privacy is dignity. It is a human right. In the domain of web browsing, the right to privacy should prevent websites from tracking user browsing activity without consent. This is the case in particular for cross-site tracking, in which website owners collude to build browsing profiles spanning multiple websites over extended periods of time. Unfortunately for users, the right to privacy conflicts with business interests. Website owners are highly interested in tracking users for the purpose of showing them ads they are more likely to click on, or to recommend products they are more likely to purchase.

We focus on the common scenario where identifying a browser is equivalent to tracking a user. The traditional way to track users is with cookies, small files that are stored by the browser at the request of the website, and forwarded to the website on demand [50]. Recent regulations restrict and supervise the acquisition of private data by websites [4, 31], and in particular require that users consent to the use of cookies. Furthermore, in an effort to protect users' privacy and curb tracking, modern browsers restrict cookie-based tracking, especially *third-party trackers* that attempt to track users across multiple unrelated websites.

To overcome the limitations of cookies, less scrupulous websites often resort to an approach called *browser fingerprinting*. To fingerprint a browser, the website provides a script that

- Conclusion位于整篇论文的最后部分
- Conclusion中，作者会概括整篇论文的工作（和abstract类似）
- 与abstract不同的是，作者在conclusion中，会更多地涉及实验的结果和整篇文章的结论

Methods/Framework-精读

(EUs), or *shader cores*, which can independently perform arithmetic and logic operations. Most consumer desktop and mobile processors from the past decade have on-chip GPUs with multiple EUs. For example, the UHD Graphics 630 GPU—integrated into Intel Core i5-8500 CPUs—includes 24 EUs, while the Mali-G72 GPU—integrated into the Samsung Exynos 9810 chipset used in Galaxy S9, S9+, Note9, and Note10 Lite devices—includes 18 EUs.

Web Graphics Library (WebGL) is a cross-platform API for rendering 3D graphics in the browser [12]. WebGL is implemented in major browsers including Safari, Chrome, Edge, and Firefox. Derived from native OpenGL ES 2.0, a library designed for developing graphic applications in C++, WebGL implements a JavaScript API for rendering graphics in an HTML5 canvas element. WebGL takes a representation of 3D objects as a list of *vertices* in space and information on how to render them, and translates them into a two-dimensional raster image that can be displayed on screen. WebGL abstracts this process as a pipeline. Two pipeline steps which are of interest to this work are the *vertex shader*, which places the vertices in the two-dimensional canvas, and the *fragment shader*, which determines the color and other properties of each fragment. The vertex and fragment shaders can run user-supplied programs, written in a C-derived programming language named *GL Shading Language* (GLSL).

III. GPU FINGERPRINTING

A. Motivation

Similar to past work [39, 52], we aim to uniquely identify devices. However, unlike previous work, which rely on the diversity of hardware and software configurations, we focus on

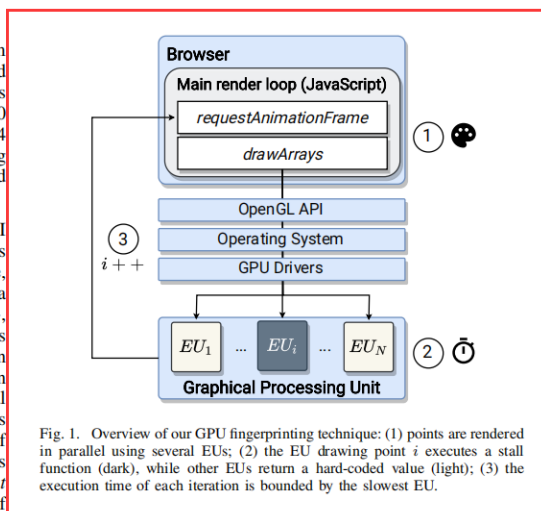


Fig. 1. Overview of our GPU fingerprinting technique: (1) points are rendered in parallel using several EUs; (2) the EU drawing point *i* executes a stall function (dark), while other EUs return a hard-coded value (light); (3) the execution time of each iteration is bounded by the slowest EU.

of the targeted EU dominates the execution time of the whole pipeline. We do so by assigning the non-targeted EUs a vertex shading program that is quick to complete, while assigning the targeted EUs tasks whose execution time is highly sensitive to the differences among individual EUs. As shown in Figure 1, our fingerprint is created by executing a sequence of drawing operations. We measure the time to draw a sequence of points with carefully chosen shader programs. The technique consists of three main steps:

- Methods/framework板块负责阐述整篇文章对于所研究问题的解决思路
- Methods板块中通常会有overview图，即将整个methods板块以流程图的方式呈现出来
- 先看流程图，粗浅了解解决思路，然后再细看methods正文部分

Results(Experiment)-精读

TABLE I. ACCURACY GAINS ACHIEVED UNDER LAB CONDITIONS

Device Type	GPU	Device Count	Timer	Base Rate (%)	Accuracy (%)	Gain
Intel i5-3470 (GEN 3 Ivy Bridge)	Intel HD Graphics 2500	10	Onscreen	10.0	93.0±0.3	9.3
			Offscreen	10.0	36.3±1.6	3.6
Intel i5-4590 (GEN 4 Haswell)	Intel HD Graphics 4600	23	Onscreen	4.3	32.7±0.3	7.6
			Offscreen	4.3	63.7±0.6	14.7
			GPU	4.3	15.2±0.5	3.5
			GPU	6.7	42.2±0.7	6.3
Intel i5-8500 (GEN 8 Coffee Lake)	Intel UHD Graphics 630	15	Offscreen	6.7	55.5±0.8	8.3
			GPU	6.7	53.5±0.8	8.0
			GPU	10.0	70.0±0.5	7.0
Intel i5-10500 (GEN 10 Comet Lake)	Nvidia GTX1650	10	Offscreen	10.0	95.8±0.9	9.6
			GPU	25.0	46.9±0.4	1.9
Apple Mac mini M1	Apple M1	4	GPU	25.0	73.1±0.7	2.9
Samsung Galaxy S8/S8+	Mali-G71 MP20	6	Onscreen	16.7	36.7±2.7	2.2
Samsung Galaxy S9/S9+	Mali-G72 MP18	6	Onscreen	16.7	54.3±5.5	3.3
Samsung Galaxy S10e/S10/S10+	Mali-G76 MP12	8	Onscreen	12.5	54.1±1.5	4.3
Samsung Galaxy S20/S20 Ultra	Mali-G77 MP11	6	Onscreen	16.7	92.7±1.8	5.6

each class, the devices were purchased through the same order, configured with our University's official operating system image, and located in the same temperature-controlled lab. The mobile devices include multiple generations of Samsung Galaxy devices, all sourced through the Samsung Remote Test Lab [10]. All the mobile devices were Android-based and featured Samsung Exynos CPUs and Mali GPUs.

Comparison With Prior Fingerprinting Techniques. Before evaluating our technique, we reproduced and tested several state-of-the-art web-based fingerprinting techniques.

value, or mode, for each of the input sizes. We reproduced the web-based variant of the method, and tested it on our GEN 4 corpus. We found that the modes did not contain any data useful for fingerprinting. This is likely because since July 2018 Chrome contains countermeasures designed to prevent fine-grain timing measurements, as part of the wider fallout of the Spectre attacks [29, 60, 62, 77]. All our measurements returned either zero or five microseconds (with some added randomness). We conclude that, currently, the method presented by Sánchez-Rola et al. is not practical.

- Results部分主要概述了论文的实验部分
- 实验部分包含两大部分：实验设置，实验结果
- 实验设置是作者展示自己如何设计实验，实验中的各种参数是什么
- 实验结果展示作者的实验的结果，同时分析实验结果，得出结论

文献笔记

我可能需要偶尔复习下这篇文章，但是我总不可能重新看一遍呀？ 😩

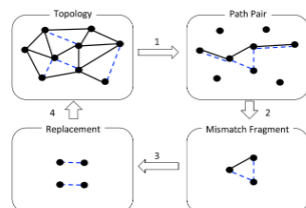
如何做文献笔记？

- 文献笔记软件：
 - Mendeley
 - EndNote
- 如何记录文献笔记：
 - 对单篇文献而言：
 - 记录大体脉络和中心思想
 - 对所有相关文献而言：
 - 按照不同分类整理好对应的文献笔记，比如按照研究问题的不同

文献笔记

方法类型: pair-matching

整个框架由以下四个步骤构成:



1. 拓拓扑构建: 收集从同一个AS到同一个目的地的数百万对traceroute路径和BGP路径

2. 定位不匹配片段 (核心工作): 定位不匹配的路径对中的不匹配的片段

3. 定位一对一AS替换: 将每个不匹配的片段转换为多个一对一的AS替换

4. 虚假链路关联: 将拓扑上的每个虚假链路与创建虚假链接的替代品相关联

定位不匹配片段 (核心工作) 的具体方法:

作者首先将AS路径转换为字符串, 基于traceroute推断出的AS路径将尽可能与BGP推断出的AS路径吻合的猜想, 将定位不匹配字符串的问题转换为了一个LCS (longest common subsequence) 问题, 并提出了系统性分析的5个步骤:

	(a) substitute	(b) end-extra	(c) unmapped	(d) tie-break	(e) loop	(f) missing tail	(g) omission
BGP path	A B C D	A B C	A B C	A B C D	A B C D	A B C	A B
Traceroute path	A E C D	A B C	A ? C	A C B D	A C A B	A C D	A * B
LCS solution	A - B + E - C - D	A - B + C - D	A - B + ? - C	A + C - B - C - D	A - C + A - B	A - B - C + D - D	A * + B
Mismatch fragment	A - B + E - C	A - B + C - D	A - B + ? - C	A - C - B - C - D	A - C + A - B	A - B - C + D - D	A * + B

Fig. 3. Examples of AS path pairs with their LCS solution and mismatch fragments. '*' and '?' are omitted if not being in mismatch fragments.

- **Special tokens:** 两个字符串匹配的结果中出现了特殊字符就必定存在路径不匹配
- **Tie-break:** 存在多种替代方式时, 使用在BGP路径中=符号出现得尽量比较早的那种
- **Loop:** 不匹配的片段是内部循环
- **Missing tail:** AS路径尾部丢失
- **Omission:** 丢弃出现的额外路径 (出现+的)

最后对不匹配片段出现的原因进行了分析

• 文献笔记的要点:

- 严简意骸, 用自己的话总结
- 不需要记录introduction内的研究背景
- 将Methods里面的主要框架记好
- 涉及到使用数据的情况需要记录好实验所采用的数据集来源

文献阅读分享 感谢倾听

王沛然