南 开 大 学

本科生毕业论文（设计）

中文题目：**基于条件概率模型的口令安全防御技术研究**

外文题目：A Study of Conditional Password Modelling in Password Security Defense

学　　号：2213218

姓　　名：张逸非

年　　级：2022 级

专　　业：计算机科学与技术

系　　别：计算机科学与技术

学　　院：计算机学院

指导教师：汪定　教授

完成日期：2026 年 5 月

关于南开大学本科生毕业论文（设计）的声明

本人郑重声明：所呈交的学位论文，是本人在指导教师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或没有公开发表的作品内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年　　月　　日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年　　月　　日

摘 要

口令是最主要的身份认证方式，在可见的未来无可替代。口令安全面临的最主要威胁是口令猜测攻击，可根据攻击场景分为两类：在线猜测攻击和离线猜测攻击。在线猜测攻击中，攻击者通过网页端或其它在线登陆界面，直接猜测用户的口令进行登录，主要特点是攻击者所能进行的猜测次数一般较为有限；离线猜测攻击中，攻击者通过本地的专用口令猜测软件和计算设备，尝试对用户的口令进行离线破解，其主要特点是攻击者所能进行的猜测次数仅受限于其所能担负的时间成本。

为了抵御口令猜测攻击这一威胁，学术界和工业界提出了一系列口令防御技术。在口令本身的抗猜测性上，学术界和工业界提倡运营商在用户注册账户时使用口令强度评估器（Password Strength Meter, PSM），对用户口令在口令猜测攻击下的抗猜测性进行评估，并及时为用户提供口令强度反馈，引导用户选用更安全的口令。在口令库对离线猜测攻击的抗猜测性上，学术界还提出使用基于蜜加密技术的蜜口令库方案，对用户储存在口令管理器等地的口令库进行蜜加密，进而提升用户口令库密文对离线主口令猜测攻击的抗猜测性。

然而，上述口令安全防御技术往往对攻击者在攻防场景中的信息不对称性缺乏考虑，鲜见对攻击者不对称信息优势的有效建模。在口令强度评估中，大多数口令强度评估器只考虑非定向口令猜测场景。实际上，攻击者可以获取用户在其它网站泄露的口令，基于用户的口令重用行为实施重用攻击，现有口令强度评估器在该猜测场景下的准确性有待系统性评估。在蜜口令库方案中，防御者所使用的口令概率模型，其建模的口令分布和真实口令库存在偏差，攻击者可以用辅助口令数据集对真实口令分布进行逼近，进而利用这一差别实施更为有效的区分攻击。如何在上述口令防御技术中，对攻击者的不对称信息加以考虑，提升口令防御技术的安全性，仍然缺乏系统性的研究和解决方案。

本文围绕口令安全防御技术中对攻击者不对称信息的建模，基于口令的条件概率模型对口令安全防御技术的安全性进行增强，主要完成了如下工作：

1. **利用口令重用模型设计了非定向的口令强度评估器**。在评估用户口令强度时，考虑用户的口令重用行为可以提高口令强度评估的准确性。口令重用模型作为条件概率模型，可以准确捕捉用户口令重用行为。然而，这类模型需要用户的旧口令作为输入，对模型进行条件化，防御者难以

在未获取用户旧口令的情况下，利用口令重用模型对用户口令强度进行评估。本文基于对用户口令重用行为的分析，指出用户不仅重用旧口令，还会重用流行口令。基于这一结论，本文成功利用现有的口令重用模型捕捉用户基于流行口令的重用行为，设计了非定向口令强度评估器。该口令强度评估器无需获取用户旧口令，只需利用流行口令词典对口令重用模型进行条件化，即可对用户口令在非定向在线猜测攻击下的强度进行有效评估。

2. **基于用户口令行为，设计了适用于多场景的口令强度评估器**。通过对大规模真实口令数据集进行统计分析，我们将用户使用弱口令的行为视为基于流行口令的重用行为，并总结了这种重用行为和基于旧口令重用行为的特点和区别。在对用户口令行为进行观察的基础上，我们设计了对口令重用模型进行连续学习的具体技术路线，从而提出 VersaPSE 多功能口令强度评估器设计框架。该框架能够通过连续学习技术，让口令重用模型同时捕捉两种口令重用行为，从而系统性地将其改造成多功能口令强度评估器。通过 VersaPSE 框架，我们将 KNNGuess、PointerGuess、Pass2Edit、PassBERT 和 Pass2Path 口令重用模型分别改造成了五款多功能口令强度评估器。实验结果表明，这五款多功能口令强度评估器在口令重用攻击、非定向口令猜测攻击两种强度评估场景中，达到了现有一系列口令强度评估器中的先进水平。
3. **概率分布、语义特征双条件化的自适应蜜口令库方案**。本文使用量化变分自编码器（VQ-VAE），设计了 VQ 自适应口令概率模型和深度学习口令重用机制，构建了深度学习自适应蜜口令库方案。该方案利用 VQ-VAE 将口令根据概率分布、语义特征，映射到离散向量上，并使用该向量条件化口令概率模型。此时，口令概率模型将会在离散向量诱导的口令子空间里对口令进行编码、解码，让采样得到的诱饵口令在分布、语义上和真实口令具有相似性，实现自适应的口令概率建模。基于分布、语义、重用特征的一系列区分攻击的安全性分析结果表明，本文提出的蜜口令库方案，在安全性上超越了学术界已有的所有蜜口令库方案。

通过以上研究内容，本文通过条件概率模型，在口令强度评估、蜜口令库方案两大口令安全防御技术中对攻击者可能拥有的不对称信息优势进行了建模，进而提升了二者在现实攻防场景中所能为用户提供的安全性。

关键词：口令安全；口令强度评估；蜜口令库方案；条件概率模型

Abstract

Passwords remain the dominant form of user authentication, and will remain irreplaceable for the foreseeable future. The primary threat to password security comes from password guessing attacks, which can be categorized into two types based on the attack scenario: online guessing attacks and offline guessing attacks. Online guessing attack is where an attacker directly guesses a user's password by interacting with the online authentication server via web browser or other authentication interfaces, often with the number of allowed password guesses strictly limited. In offline guessing attacks, an attacker employs dedicated password guessing software as well as hardware to crack a user's password locally, with time and computational resources the attacker can afford as the only constraint.

To mitigate the threat of password guessing attacks, both academia and industry have proposed a variety of password security defense techniques. On the front of password strength, almost every respectable service deploys password strength meters (PSMs) during user registration, which estimate the guessability of user-chosen passwords against password guessing attacks and provide timely strength feedback, thereby guiding users toward choosing stronger passwords. On the front of protecting password vaults against offline master password guessing attacks, researchers have proposed honey vault schemes based on honey encryption, which apply honey encryption to password vaults stored in password managers, thereby enhancing the resistance of encrypted password vaults to such offline guessing attacks.

However, the above password defense techniques often overlook the asymmetric knowledge between attackers and defenders in real-world scenarios, and rarely provide effective modeling of the attacker's advantage caused by such asymmetry. In password strength evaluation, most PSMs only consider the untargeted password guessing scenario. In practice, an attacker can obtain passwords leaked from a user's other accounts and launch credential tweaking (password reuse) attacks by exploiting the user's password reuse behavior; the accuracy of existing PSMs under this guessing scenario still lacks systematic evaluation. In honey vault schemes, the password distribution modeled by the defender's probabilistic password model inevitably deviates from that of

real password vaults; an attacker can leverage auxiliary password datasets to better approximate the true password distribution, thereby exploiting this discrepancy to mount more effective distinguishing attacks. How to systematically account for the attacker's asymmetric knowledge in these password defense techniques, so as to enhance their security, still lacks systematic research and comprehensive solutions.

This thesis centers on modeling the attacker's asymmetric knowledge in password security defense techniques and enhancing the security of these techniques through conditioned probability modelling of passwords. The main contributions of this thesis are as follows:

1. **An untargeted password strength meter using password reuse models.** Considering users' password reuse behavior can improve the accuracy of password strength evaluation. Password reuse models, as conditional probability models, are capable of accurately capturing password reuse behavior. However, these models require a user's old password as input to condition the model, making it difficult for defenders to employ such models without access to users' prior passwords. Based on an analysis of password reuse behavior, this thesis shows that users reuse not only their own old passwords, but also popular passwords. Building on this finding, we successfully leverage existing password reuse models to capture users' password reuse behavior towards popular passwords, and design an untargeted password strength meter. This meter requires no access to users' old passwords—it only needs a dictionary of popular passwords to condition the password reuse model, and can then effectively evaluate the strength of user passwords under untargeted online guessing attacks.
2. **A multi-purpose password strength meter based on user password behavior.** Through statistical analysis of large-scale real-world password datasets, we recast users' behavior of choosing weak passwords as popular-password reuse behavior, and characterize the properties of and differences between this reuse behavior and old-password reuse behavior. Based on these empirical observations, we design a concrete continual learning methodology for password reuse models, and propose VersaPSE, a versatile password strength evaluation framework. This framework applies continual learning to enable a password reuse model to capture the two types of reuse behavior simultaneously, thereby

systematically transforming it into a multi-purpose password strength meter. Through the VersaPSE framework, we have respectively transformed the KNNGuess, PointerGuess, Pass2Edit, PassBERT, and Pass2Path password reuse models into five multi-purpose password strength meters. In our experimental analysis, results, all five multi-purpose PSMs achieve state-of-the-art accuracy under both credential tweaking attacks and untargeted password guessing attacks, compared with existing PSM approaches.

3. **A dual-conditioned adaptive honey password vault scheme based on probability distribution and semantic features.** We employ vector-quantized variational autoencoders (VQ-VAEs) to design a VQ-adaptive password probability model and a deep-learning-based password reuse mechanism, constructing an adaptive deep-learning honey password vault scheme. This scheme uses VQ-VAEs to map each password, according to its probability distribution and semantic features, onto discrete latent vectors, and then uses these vectors to condition a password probability model. In this way, the password probability model encodes and decodes passwords within the password subspaces induced by the discrete vectors, so that the sampled decoy passwords resemble the real passwords in both distribution and semantics, thereby realizing adaptive password probability modeling. Security evaluation results under a series of distinguishing attacks based on distribution, semantic, and reuse features show that the proposed honey password vault scheme outperforms all five existing academic schemes in terms of security.

The above contributions leverage conditional probability models in password security defense techniques, specifically password strength evaluation and honey vault schemes, thereby enabling the modeling of the attacker's asymmetric knowledge, improving the security that these techniques can provide in real-world scenarios.

Key Words: password security; password strength evaluation; honey vault schemes; conditional probability model

目 录

摘要 …… I
Abstract …… III
目录 …… VI
第一章 引言 …… 1
第一节 研究背景 …… 1
第二节 研究现状 …… 3
1.2.1 口令强度评估 …… 3
1.2.2 蜜加密技术与蜜口令库方案 …… 5
第三节 本文工作 …… 6
第四节 本文结构 …… 7
第二章 基于口令重用行为的非定向口令强度评估器 …… 8
第一节 研究背景 …… 8
2.1.1 研究动机 …… 9
2.1.2 本章贡献 …… 11
第二节 相关工作 …… 12
第三节 预备知识 …… 13
2.3.1 口令数据集 …… 13
2.3.2 Pass2Edit 口令重用模型 …… 14
2.3.3 评价指标 …… 16
2.3.4 本章比较的现有非定向口令强度评估器 …… 17
第四节 基于流行口令的口令重用行为 …… 18
第五节 基于 Pass2Edit 的非定向口令强度评估器 …… 19
第六节 实验设计与结果 …… 20
2.6.1 实验场景设定 …… 20
2.6.2 实验结果 …… 22
第七节 本章小结 …… 23
第三章 多功能口令强度评估器的设计 …… 26
第一节 研究背景 …… 26
3.1.1 研究动机 …… 27

3.1.2 本章贡献 ····· 28
第二节 相关工作 ····· 29
3.2.1 口令重用模型 ····· 29
3.2.2 口令重用攻击下的口令强度评估 ····· 32
第三节 预备知识 ····· 32
3.3.1 口令数据集 ····· 32
3.3.2 口令重用数据集的构造 ····· 33
3.3.3 持续学习技术 ····· 34
3.3.4 评价指标 ····· 34
第四节 基于持续学习的多功能口令强度评估器设计框架 ····· 35
3.4.1 两种口令重用行为的特点 ····· 35
3.4.2 持续学习关键设计细节的理论分析 ····· 37
3.4.3 用持续学习改造口令重用模型 ····· 41
第五节 实验设计与结果 ····· 44
3.5.1 各口令重用模型的具体改造方案 ····· 44
3.5.2 口令重用攻击下的口令强度评估 ····· 45
3.5.3 非定向口令猜测攻击下的口令强度评估 ····· 46
第六节 本章小结 ····· 47
第四章 深度学习技术在蜜口令库方案中的应用 ····· 49
第一节 研究背景 ····· 49
4.1.1 研究动机 ····· 50
4.1.2 本章贡献 ····· 51
第二节 相关工作 ····· 52
第三节 预备知识 ····· 54
4.3.1 蜜加密技术与蜜口令库方案 ····· 54
4.3.2 区分攻击 ····· 55
4.3.3 文本分类模型 ····· 58
4.3.4 口令概率模型 ····· 59
4.3.5 量化变分自编码器 ····· 60
4.3.6 自适应层归一化 ····· 61
4.3.7 口令数据集 ····· 61
第四节 基于子空间划分的自适应蜜口令库方案 ····· 62
第五节 基于 VQ-VAE 的自适应口令概率模型 ····· 64

4.5.1 利用 VQ-VAE 进行口令空间切分 ····· 65
4.5.2 口令概率分布的 VQ-VAE 空间切分 ····· 65
4.5.3 口令语义特征的 VQ-VAE 空间切分 ····· 67
4.5.4 自适应口令概率模型 ····· 69
第六节 基于 PointerGuess 的口令重用机制 ····· 74
4.6.1 口令重用模型的筛选标准 ····· 74
4.6.2 逐字符的重用口令编码方法 ····· 76
4.6.3 基于条件概率的口令重用判定方法 ····· 77
第七节 实验设计与结果 ····· 79
4.7.1 贴合实际攻防场景的实验设计 ····· 79
4.7.2 分布区分攻击下的安全性分析 ····· 80
4.7.3 语义区分攻击下的安全性分析 ····· 82
4.7.4 重用区分攻击下的安全性分析 ····· 84
4.7.5 现有区分攻击下的安全性分析 ····· 86
第八节 性能分析 ····· 87
第九节 本章小结 ····· 88
第五章 总结与展望 ····· 90
第一节 本文工作总结 ····· 90
第二节 本文工作的局限性 ····· 91
第三节 未来工作展望 ····· 91
附 录 ····· 93
第一节 部分非定向口令强度评估器的设定 ····· 93
第二节 部分重用口令强度评估器的设定 ····· 93
第三节 用更多相似度指标对用户口令重用行为进行统计 ····· 94
第四节 用其它指标评价重用口令强度评估器的准确性 ····· 95
第五节 其它多功能口令强度评估器的 *WSpearman* 热力图 ··· 96
参考文献 ····· 98
致 谢 ····· CVII
本科期间的主要工作和成果 ····· CIX
个人简历 ····· CX

第一章　引言

第一节　研究背景

互联网不仅在不断发展，而且已经无可争议地成为了绝大多数人生活中不可或缺的部分。在 2025 年，调研显示全世界已有 60 亿网民[1]，其中我国有 11 亿网民[2]，占总人口的百分之 75 以上。互联网的普及和发展，在为人们提供极大的便利之余，也让网络空间安全问题成为了国家安全的“第五疆域”。没有网络空间安全就没有国家安全，保障包括用户账户和隐私安全在内的网络空间安全，成为了保障国家安全的必要条件。

口令作为用户账户和隐私安全的第一道防线，其安全性直接关系到用户账户的安全性。调查显示，在 2020 年，平均每个网民拥有 90 个互联网账户，而到了 2024 年，每个用户平均拥有的账户则快速增长到了 100 至 150 个[3]。用户账户数量的增加，导致了用户需要记忆更多的口令，进而导致了用户口令安全性的下降。调查显示，在 2025 年，60% 以上的用户存在口令重用行为 [4]，其中 24% 的用户甚至将完全相同的口令用在不同账号上[5]；在 2024 年，高达 39% 的用户使用弱口令（如“123456”）[6]。这些脆弱的口令行为，在口令泄露事件越来越频繁的今天[7–9]，直接导致了大量用户账户安全性的下降，造成了一系列经济损失，成为了网络空间安全的重要威胁之一。例如，在 2025 年，苹果、脸书、Snapchat 等多个网站的口令泄露事件导致了十亿用户的账户和口令被泄露[10]；在 2021 年，LastPass 口令管理器的云端口令库密文遭到泄漏，攻击者通过离线猜测恢复得到的明文口令库，窃取了至少三千五百万美元的加密货币等其它资产，给用户造成了巨大经济损失[11, 12]。研究如何提升用户口令安全性，并抵御针对口令的一系列攻击，成为了网络空间安全领域的重要研究方向之一。

口令所面临的主要安全威胁之一是口令猜测攻击[13–20]，即攻击者通过尝试不同的口令来猜测用户口令、破解用户账户的攻击方式。口令猜测攻击可以分为在线猜测攻击和离线猜测攻击两大类[21]，前者指攻击者直接在登录界面进行口令猜测[22]，后者指攻击者获取到用户账户的口令哈希值后，在本地进行口令猜测。无论是在线猜测攻击还是离线猜测攻击中，用户所选口令的强度均显著影响攻击成功率[21, 23]。用户选择的口令越弱，攻击者就越容易通过口令猜测攻击来破解用户账户；反之，用户选择的口令越强，攻击者就越难以通过口令猜测攻击来破解用户账户。因此，提升用户口令安全性的一个重要方法，就是提

升用户所选口令的强度。

为了提升口令在猜测攻击下的抗猜测性，口令强度评估器能够对用户所选口令的强度进行评估，并将评估结果反馈给用户，帮助用户了解所选口令的安全性，从而引导用户选择更安全的口令[21, 23–25]。近年来，学术界和工业界设计了一系列类型多样、技术路线各异的口令强度评估器，有的口令强度评估器在诸如微软、12306、Skype 等网站得到部署，可以影响上千万用户的口令选择[26–29]。然而，研究表明，现有的口令强度评估器在准确性、结果一致性等各个方面良莠不齐：只有准确的口令强度评估器才可以有效引导用户选择强度更高的口令，而不准确、反馈不一致的口令强度评估器，反而有可能让用户对口令强度的认知产生偏差，甚至误导用户选择看似安全实则脆弱的口令[21, 24]。设计准确有效的口令强度评估器，是提升用户口令安全性的一个重要前提。

通过口令强度评估器对用户进行引导，进而提高口令本身的强度，是从用户行为的视角出发，对口令安全进行保护。然而，每年层出不穷的口令文件泄露事件[10, 30–32]，使得攻击者可以在近乎不受猜测数目限制的情况下，用先进的计算设备对口令进行离线破解。在离线的口令猜测场景下，单从用户视角出发保护口令安全显得捉襟见肘。从口令本身的存储、加密机制出发，从服务提供方的角度提升口令安全，成为了保护口令安全的另一个必要条件。

其中，从服务提供方角度提升口令安全的一种重要方式，是使用蜜口令库方案[33–36]，对用户的口令库（如口令管理器中存储的一系列口令）进行基于主口令的蜜加密。相比于传统的加密方案，蜜口令库的好处是，即便发生数据泄露事件导致攻击者获取加密后的口令库密文，攻击者在对主口令进行离线猜测攻击时，仍会解密得到一系列似是而非的诱饵口令库。具体来说，任意攻击者用于解密的主口令猜测，不论猜测是否正确，都能在蜜口令库方案下解密得到具有语义信息、符合口令分布的明文口令库。这样一来，攻击者在猜测攻击中将得到一系列诱饵口令库。真实口令库和诱饵口令库真真假假、虚实难辨，攻击者在理想情况下无法将二者进行有效区分，只能对得到的明文口令逐一进行在线验证。此时，攻击者不受猜测次数限制的离线猜测攻击能力，就被转化为了猜测次数高度受限、防御机制更加完善的在线猜测攻击能力。

蜜口令库方案的有效性在于其产生的诱饵口令库是否足够真实，难以被攻击者鉴别出来。不幸的是，现有研究表明，目前学术界提出的一系列蜜口令库方案，在诱饵口令库的难区分性上均存在缺陷[35, 37, 38]。例如，Chatterjee 等人在 2015 年提出的 NoCrack 蜜口令库方案，其诱饵口令库在口令分布上与真实口

令库存在显著差别，攻击者可以通过口令分布轻松将诱饵口令库区分出来，使NoCrack的安全性降低了至少50%[35]。再比如，Golla等人提出的蜜口令库方案虽然在口令分布上相对安全，但是其分布自适应机制间接泄露了真实口令库的分布信息，使攻击者能够利用这一脆弱性排除掉大量的诱饵口令库[36, 37]。如何对现有蜜口令库方案存在的脆弱性进行系统的分析，并通过总结其存在各方面脆弱性的原因，设计一款真正安全、诱饵口令库真正难以被攻击者鉴别的蜜口令库方案，是一个亟待解决的问题。

第二节　研究现状

1.2.1　口令强度评估

口令强度评估的本质是，利用口令强度评估器，在特定的口令猜测攻击场景下，对用户口令抵抗该口令猜测攻击的抗猜测性进行估计。本文主要关注两类口令猜测攻击下的口令强度评估：非定向口令猜测攻击和口令重用攻击。

非定向猜测攻击下的口令强度评估

非定向猜测攻击指攻击者不针对某个特定的用户，也不利用关于某特定用户的先验知识（用户名、姓名、旧口令等），直接对整个目标身份验证系统里的所有用户进行口令猜测攻击[15, 21, 39]。此时，攻击者的目标并不是提升对某一特定用户口令的破解成功率，而是力求在攻击中提升所能破解的用户口令总数。目前为止，学术界已经提出了一系列先进的非定向口令猜测模型，如PCFG[40]、wang2023rfguess[13]、RankGuess[15]、PassLLM[41]等。随着攻击者用于进行非定向猜测攻击的口令猜测模型越来越先进，硬件设备的性能越来越强，利用口令强度评估器引导用户选择在非定向猜测攻击下更加安全的口令，成为了主流网站和服务提供商的共识[21, 24, 27–29, 42]。

工业界所提出的非定向口令强度评估器，可以大致分为基于模式和基于启发式方法的两类。其中，基于模式的口令强度评估器以12306-PSM和Microsoft-PSM为代表，其根据用户口令长度、所包含的字符种类等口令所展现出的模式特征，对用户口令在非定向猜测攻击下的强度进行评估[26, 28]。这类口令强度评估器的设计方案非常简单，部署也较为便捷，收到了较多服务提供商的青睐。基于启发式方法的口令强度评估器，以Zxcvbn-PSM为代表，则在设计原理和评估指标上更加复杂[43]。这类口令强度评估器往往利用一系列启发式规则，例如口令中出现的单词和字段、预存储的弱口令黑名单，来对用户口令的

强度进行更加系统全面的估计。研究表明，Zxcvbn-PSM 这类基于启发式方法的口令强度评估器准确性较好，被一系列网站所采用和部署[24, 29, 44]。

学术界所提出的非定向口令强度评估器则更加重视在设计中采纳口令猜测领域的前沿技术。这些口令强度评估器往往由口令猜测模型改造而来，进而得以从攻击者的视角出发，直接或间接估计用户口令在该口令猜测模型下的抗猜测性[45, 46]。例如，fuzzyPSM[23] 由经典的口令猜测模型 PCFG[40] 改造而来，在口令猜测模型中结合了用户的口令重用行为，成为了迄今为止最为准确的非定向口令强度评估器之一[21, 24]。此外，学术界还提出了一系列基于其它各类口令猜测模型的口令强度评估器，例如 RNN-PSM[39]、Markov-PSM[46] 等，同样实现了较为迅速、准确的口令强度评估。

值得一提的是，现有研究指出，即便在攻击者不使用旧口令等额外信息的非定向口令猜测场景中，考虑口令重用行为依然有助于提升口令强度评估的准确性[23]。事实上，fuzzyPSM——迄今为止最为成功的非定向口令强度评估器之一，就在非定向口令强度评估中，考虑了用户在数据集层面上的宏观口令重用行为，进而实现了比其它口令强度评估器更高的准确性[21, 24]。然而，fuzzyPSM 对口令重用行为的建模高度依赖启发式重用特征，如何在非定向口令强度评估中不受启发式重用机制的限制，更有效地利用口令重用行为，依然是一个有待解决的问题。

口令重用攻击下的口令强度评估

口令重用攻击指的是，攻击者利用口令重用行为，基于用户在一些网站已经泄露的账户口令，对该用户在其它网站使用的口令进行攻击。攻击者可以将其获取的用户其他账户口令，原封不动或稍加修改，然后用于攻击该用户的其它账户。研究表明，攻击者只需获取用户的单个旧口令，即可在低至 1000 次猜测内，达到 70% 以上的破解成功率[22, 47, 48]。截止目前，层出不穷的口令泄露事件已经导致了数十亿用户的口令遭到泄漏[9, 10, 30, 31]，可以被攻击者轻易获取，为攻击者实施口令重用攻击提供了极其丰富的材料。上述事实说明，口令重用攻击已经对用户的口令安全产生了日渐严重的现实威胁。

为了提升用户口令在口令重用攻击下的抗猜测性，学术界提出了基于重用的口令强度评估器，并初步验证了其有效性。2019 年，Pal 等人提出基于文本相似度的 Vec-PPSM 口令强度评估器，声称其能够有效检测出重用攻击场景中 90% 以上的脆弱口令[47]；2023 年，Xu 等人基于 PassBERT 口令重用模型，设计

了 BERT-PSM 口令强度评估器，能够通过模拟口令重用攻击来评价用户口令强度[14]；2024 年，Xiu 等人在利用口令重用模型模拟重用攻击的基础上，进一步引入一系列启发式方法，实现了更准确的口令强度评估[48]。不过，据我们所知，迄今为止，尚未有人系统地对这些基于重用的口令强度评估器进行准确性评价。现有的基于重用的口令强度评估器，其在现实攻击场景中的准确性和实用性，仍然有待进一步检验。

1.2.2 蜜加密技术与蜜口令库方案

蜜加密技术是一种抵御离线密钥猜测攻击的加密技术。在使用蜜加密技术对消息空间中的一个明文消息进行加密得到密文之后，对于任意密钥，不论其正确与否，都能够按照消息空间里消息的概率分布，从密文中解密得到消息空间中由密钥唯一确定的明文消息。此时，攻击者在猜测次数无限制的离线密钥猜测攻击里，无法通过解密结果直接确定密钥猜测是否正确，必须通过在线验证才能对真实消息和错误消息进行区分。这样一来，没有猜测次数限制的攻击者离线猜测能力就被弱化为了有限制的在线验证能力，提升了密文对离线猜测攻击的抵御能力。

蜜口令库方案是蜜加密技术在口令安全领域的应用。口令本身是低熵密钥，其作为消息在消息空间中的概率分布可以较好用口令概率模型进行建模。这就让使用蜜加密技术保护用户口令库成为了现实可行的方案。具体来说，用户选定了用于加密口令库的主口令之后，蜜口令库方案通过口令概率模型，按照概率将用户口令库中的每个口令映射到编码空间（Code Space）中，拼接得到一个比特串。随后，使用主口令对该比特串进行加密，即可得到蜜加密后的口令库密文。攻击者即便通过攻击云服务器等方式，截获了用户口令库密文，其在对主口令进行离线猜测的过程中，不论其猜测的主口令是否正确，都会通过口令概率模型反向得到似是而非、真假难辨的明文口令库。这样一来，诱饵口令库和真实口令库混杂在一起，哪怕攻击者猜中了主口令，也无法确认哪次猜测的主口令是正确的、哪些猜测是错误的，必须逐一将得到的明文口令提交到对应的在线身份认证平台上。由于在线登录次数有限，攻击者原本不受猜测数目限制的离线口令猜测能力就被削弱成了猜测次数高度受限的在线口令猜测能力。

不难看出，上述蜜口令库方案的安全性，归根结底取决于诱饵口令库与用户真实口令库之间的不可区分性。然而，一系列研究表明，现有的蜜口令库方案普遍难以抵御精心设计的区分攻击。例如，基于口令分布的 KL 散度攻击、单

口令攻击和理论最优攻击[35, 36, 38]，能够将学术界 Chatterjee 等人提出的 NoCrack 方案[34]、Golla 等人提出的自适应方案[35] 在安全性上削弱将近 50%；基于口令重用特征的相似度攻击，则同样能将这两种方案的安全性削弱 40% 以上[36, 49]。这些区分攻击之所以有效，归根结底是因为防御者训练得到的口令概率模型，从其中采样得到的诱饵口令在语义、概率分布等各个方面和用户的真实口令存在不同程度的偏差[35, 49]。攻击者可以利用这种偏差，并且通过获取比防御者更贴近真实口令分布的辅助口令数据集，来根据不对称的信息实施更为有效的区分攻击[49]。对于防御者来说，如何在设计蜜口令库方案时，考虑诱饵口令和真实口令所存在的偏差，并且提前对其进行建模，提升蜜口令库方案的安全性，是一个颇具挑战性的研究课题。

第三节　本文工作

围绕条件化的口令建模方法在口令安全防御技术中的应用，本文在口令强度评估、蜜口令库方案两个方面完成了以下三个工作：

- **流行口令条件化的非定向口令强度评估器**。本文在对大量真实口令数据集中所体现的口令重用行为进行了基于统计的分析，发现相当一部分用户的口令和流行口令是高度相似的。这说明用户不仅重用旧口令，而且会重用流行口令。在这一发现的基础上，本文利用 Pass2Edit 口令重用模型，使其学习用户在流行口令上的重用行为，提出了基于口令重用行为的非定向口令强度评估器 EditPSM。该评估器用一个与待评估口令相似的流行口令，对重用模型进行条件化，然后计算得到待评估口令在模型中的条件概率，概率越高则口令抗猜测性越低。实验结果表明，本文提出的 EditPSM 口令强度评估器在准确性上超越了现有的一系列口令强度评估器，首次成功将口令重用模型应用到非定向口令强度评估中。
- **基于条件概率的多功能口令强度评估框架**。本文利用持续学习技术，在基于口令重用模型的非定向口令强度评估器的基础上，构建了多功能口令强度评估框架 VersaPSE。该框架能够让现有的一系列口令重用模型，通过连续学习技术，建模基于流行口令和旧口令的两类重用行为，进而通过流行口令、旧口令诱导的两类条件概率空间，同时评价口令在非定向口令猜测攻击和口令重用攻击两个场景下的抗猜测性。本文利用 VersaPSE 成功将学术界已有的五款重用模型系统性地改造为了多功能口令强度评估器，其在非定向评估、重用评估两个场景中的准确性，均超越了现有的一系列口

令强度评估器。

- **深度学习自适应蜜口令库方案**。本文从蜜口令库中真实口令库和诱饵口令库在语义、分布特征的差异出发，提出了基于子空间划分的自适应机制。具体来说，本文利用量化变分自编码器（VQ-VAE），将口令空间中具有相似特征的口令，映射到相同的高维空间离散向量上。这时，使用这些离散向量诱导的口令子空间对口令概率模型进行条件化，即可实现按子空间对口令库进行编码、解码，让诱饵口令库在概率分布、语义特征上更加相近。该深度学习自适应蜜口令库方案在安全性上超越了学术界现有的一系列方案，有效地提升了蜜口令库方案在区分攻击下的安全性。

第四节　本文结构

本文余下部分的章节安排和结构如下：

- 第二章介绍如何利用口令重用模型构建非定向口令强度评估器，并在 Pass2Edit 重用模型的基础上提出了 EditPSM，对其准确性进行了实验分析。
- 第三章在第二章的基础上，进一步利用持续学习技术构建了 VersaPSE 多功能口令强度评估框架，并将一系列现有的口令重用模型改造成了多功能口令强度评估器。
- 第四章主要介绍如何通过量化变分自编码器（VQ-VAE）对口令空间进行概率分布、语义特征两个层面的切分，并从子空间里采样诱饵口令。通过将诱饵口令的采样空间限制在与真实口令相同的子空间内，该自适应蜜口令库方案的诱饵口令在语义、概率分布上和真实口令相对一致，在各类区分攻击下实现了比现有一系列方案更高的安全性。
- 第五章对本文的工作进行了总结，并且对未来工作和有研究价值的其它问题进行了展望。

第二章　基于口令重用行为的非定向口令强度评估器

如何基于用户行为对口令强度进行建模，是口令强度评估的研究方向之一。本章主要介绍如何利用口令重用行为对非定向猜测条件下的口令强度进行建模，进而利用能捕捉口令重用行为的口令重用模型构建非定向口令强度评估器。该部分内容将为后续的第三章内容奠定基础并提供必要前提。

第一节　研究背景

从用户行为出发研究口令安全，是口令安全领域的基本方法之一。基于用户的口令行为对口令强度进行建模，进而设计更加准确的口令强度评估器，也是学术界和工业界大量口令强度相关研究的出发点。例如，相当一部分用户倾向于在口令中连续使用键盘上位置相近的字符（如“qazwsx”、“qwerty”），也有一些用户会在口令中使用网站名称（如“weibo.com”、“tianya”）。这类在用户群体中普遍存在，且高度可预测的用户脆弱口令行为，是大量用户口令强度较低，且容易被攻击者在猜测攻击中破解的成因之一，同时也是口令强度评估器经常利用的评测指标。例如，工业界广泛使用、且被一些学术界研究者认为安全可靠的 Zxcvbn 口令强度评估器，就利用了一系列用户行为作为评价口令强度的标准。学术界提出的一些数据驱动口令强度评估器[25]，也使用了上十个精心构造的启发式方法来检测用户口令中展现出的脆弱口令行为。

除了用户行为，从攻击者的视角出发来设计口令强度评估器也是常见的思路。口令强度本质上是口令在各类口令猜测攻击下的抗猜测性，如果能够模拟攻击者实施的口令猜测攻击，就可以较为准确地得到口令的抗猜测性，进而对口令的强度给出估计。例如，D’Amico 等人提出的蒙特卡洛猜测数估计方法[45]，通过对口令猜测模型的概率输出进行蒙特卡洛采样，基于口令猜测模型为口令赋予的概率，对口令被该猜测模型破解所需的猜测数目进行准确估计，并利用该猜测数作为口令强度的表征。无独有偶，学术界后续提出的一系列口令强度评估器，如 Markov-PSM、PCFG-PSM、RNN-PSM，都是基于口令猜测模型设计的[39, 40, 46, 50]。

不难看出，设计准确有效的口令强度评估器，需要准确把握用户的脆弱口令行为、攻击者的视角这两个方面。脆弱的口令行为导致了口令本身存在脆弱性，容易被攻击者的猜测攻击所破解；攻击者为了让口令猜测攻击更有效，必须考虑用户脆弱的口令行为并加以利用。例如，大量用户倾向于使用流行口令

（如“123456”），该流行口令在各类猜测攻击下均非常脆弱，是影响口令安全的重要负面因素之一[6]；反过来，各类口令猜测攻击（如基于字典的拖网猜测攻击优先利用流行口令进行猜测）和口令猜测模型（如定向口令猜测模型显式利用流行口令辅助进行猜测），也会尽一切努力来捕捉用户使用流行口令的脆弱口令行为，进而提高口令猜测攻击的成功率[22, 39, 51, 52]。

从用户脆弱口令行为的危害和攻击者视角来看，用户的口令重用行为，是对口令安全性造成最重大影响的口令行为之一。口令重用，即用户将旧的已有口令，原封不动地或稍加修改后，在新的账户上进行使用的做法，是在用户群体中普遍存在的一种口令行为。调查显示，有超过 70% 的用户拥有口令重用行为，其中将近 20% 的用户甚至将完全相同的口令用在多个不同账号上[4, 5, 53]。这种口令行为之所以脆弱，是因为它能够让用户单个账户的口令强度不再完全依赖于口令本身，而是会进一步依赖于其它口令（如该用户其它账户的口令）；一旦用户所用重用的其它口令被攻破（如该用户其他账户的口令遭到泄漏），且攻击者获取到了这些口令，那么攻击者就可以在这些口令的基础上进行基于口令重用的定向猜测攻击，简称口令重用攻击。这种攻击的威胁是巨大的，大量研究发现口令重用的攻击在一千次猜测以内的成功率可以高达 70%[22, 48]，即便在猜测次数高度受限的在线猜测场景中（如直接在登录界面尝试口令猜测）也颇具威胁；这种攻击的威胁同样是现实的，近年来海量的口令泄露事件和暗网中大量的口令数据交易，已经证明数十亿用户的口令已经遭到泄漏，并且可以被攻击者在暗网轻易获取[30–32, 54]。可以看出，基于口令重用行为，从用户行为和攻击者视角两个方面出发设计口令强度评估器，是十分有意义的。

2.1.1 研究动机

口令重用行为是非常脆弱的口令行为[22, 51, 53]，其作为口令重用攻击的基本攻击前提和利用对象，直接导致了严重现实威胁[55–57]。然而，口令重用行为在口令强度评估中的应用则困难重重。从脆弱口令行为这一方面来看，口令重用本身是一种高度复杂的行为，用户重用口令的方式、形式是非常多样的[22, 53]。例如，用户可以采用首字母大写（将”zhang123” 修改为”Zhang123”）这种简单的重用方式，也可以采用插入新字段的较复杂重用方式（将”zhang123” 修改为”zhang123nankai”），还可以采用修改字段、变换顺序的更为复杂重用方式（将”zhang123” 修改为”123nankaizhang”）。从攻击者的视角来看，攻击者为了捕捉复杂的口令重用行为，在口令重用攻击中所使用的口令重用模型也较为复杂，

若用于构建口令强度评估器将带来巨大的算力和存储负担。例如，学术界最为经典的口令重用模型之一 Targuess-II，定义了一系列启发式口令重用方式，并基于统计学模型对重用行为进行建模，模型体积高达 2G[22, 51]；目前最为先进的口令重用模型之一 KNNGuess[58]，其体积至少为 2GB（随训练集变大而继续增大），且需要 GPU 作为推理设备。此外，基于口令重用行为进行强度评估，需要获取用户所重用的口令（例如用户其它账户的口令），而这些口令属于用户隐私信息，安全管理员难以直接获取。上述种种实际困难，导致基于口令重用行为的口令强度评估器难以设计、难以发展、难以落地。

基于口令重用行为设计口令强度评估器，存在种种困难，其要害就是口令重用模型本身的可用性问题。口令重用行为本身复杂，难以准确建模，所以攻击者用于捕捉重用行为的口令重用模型较为复杂，对存储空间和算力的要求很高[58]；口令重用行为的实施对象是用户所重用的口令（如用户其它账户的口令），攻击者所使用的口令重用模型则依赖于这些作为用户隐私信息的被重用口令[22, 53]。在口令重用模型的存储和算力成本上，攻击者为了获取利益，拥有支付这些成本的动机，而普通用户则不一定愿意付出相应成本进行口令强度评估[59]；在口令重用模型所需的隐私信息上，攻击者同样可以从暗网等各种渠道来获取这些隐私信息（如用户在其它网站上泄露的口令），而安全管理员、服务提供商则受到种种限制，难以获取用户的隐私信息[60]。攻击者和防御者、用户在成本支付意愿、信息获取能力上的不对称性，造成了口令重用模型在用于构建口令强度评估器时产生的可用性问题。

目前，学术界有较多将口令重用模型用于设计口令强度评估器的尝试，如利用 PassBERT 口令重用模型构建 BERT-PSM[14]。这些研究无一例外地使用口令重用模型模拟攻击者的猜测攻击，将攻击结果（攻击者实际所需采用的猜测数）直接地或结合启发式方法进行变换后，作为口令强度评估的指标反馈给用户[14, 48, 58]。毫不意外地，这种利用口令重用模型构建强度评估器的方法，很大程度上受到了前文所述的可用性限制。以 KNNGuess 口令重用模型构建的 KNN-PSM 为例[58]，其在评估用户口令强度时需要生成一个猜测列表（即按攻击者攻击方式进行猜测生成），然后通过查表得到用户口令是否被破解、何时会被破解。这个评估过程实际上使用攻击模型完成了一次完整的猜测攻击，其需要至少 2GB 的存储空间，即便在 NVIDIA RTX 3090 GPU 这类专业计算设备上也需要运行至少一秒（而用户所能接受的实时反馈时间往往在毫秒级）。解决口令重用模型在用于评估口令强度时所面临的可用性缺陷，是将口令重用行为引入

口令强度评估的一个重要前提。

2.1.2 本章贡献

本章基于对口令重用行为的观察，利用口令重用模型构建口令强度评估模型，主要做出了以下贡献：

- **拓宽了口令重用行为**。本章对 10 个大规模口令数据集进行统计，得到用户不仅重用其它账户的口令，而且还会重用流行口令的结论。这一结论拓宽了口令重用行为的边界，使其不再局限于用户对自身旧口令的重用。同时，这一发现在使用弱口令和重用口令两种脆弱口令之间建立了桥梁，将使用弱口令视为重用流行口令，让使用口令重用模型评估非定向猜测下的口令强度成为可能。由于流行口令相对用户旧口令而言，更容易被安全管理员和服务提供商获取，该发现同样解决了“口令重用模型必须使用旧口令这一用户隐私数据”的可用性问题，为口令重用模型在口令强度评估中的实际应用提供了良好前提。
- **提出使用条件概率作为口令强度表征**。本章首次提出使用口令重用模型为用户口令分配的条件概率来表征口令强度。口令重用模型的本质上是对口令的条件概率进行建模的模型，而在口令重用攻击中，攻击者按照条件概率降序生成口令并进行猜测攻击。自然地，条件概率就是口令抗猜测性的表征，口令的条件概率越高，攻击者越有可能在口令重用攻击中将其攻破，即该口令在口令重用攻击下的强度越低。使用条件概率作为强度表征的优势是，计算条件概率无需耗费大量算力和时间生成上千个口令猜测，只需对待评估口令计算一次概率即可，所需成本与生成单个猜测相近（耗时往往在毫秒级）。同时，条件概率能够提供比猜测数更细粒度的口令强度表征，可以为用户提供更精准更直观的口令强度反馈。
- **设计基于口令重用模型的非定向口令强度评估模型**。本章基于 Pass2Edit 口令重用模型设计了 EditPSM 非定向口令强度评估器。Pass2Edit 口令重用模型本身体积较小，且能较精确地建模口令重用行为，在口令重用攻击任务中表现良好。本章在重用流行口令的口令重用行为基础上，在不获取用户旧口令的前提下，利用口令字典构造了用于训练 Pass2Edit 模型的训练集。该训练集能够让 Pass2Edit 捕捉用户重用流行口令的口令重用行为，进而让 Pass2Edit 所构建的 EditPSM 口令强度评估器在不使用旧口令的情况下，利用流行口令计算用户口令的条件概率，评估口令在非定向猜测攻

击下的口令强度。

- **大量实验**。为证实 EditPSM 在评估非定向口令猜测攻击下的口令强度时的准确性，本章基于 10 个大规模真实世界口令数据集和 8 个学术界、工业界现有的代表性口令强度评估器，进行了大量的实验。实验结果表明，EditPSM 能够快速有效地评估用户口令在非定向攻击下的口令强度，其准确性达到了现有口令强度评估器的先进水平。这一结果说明本章所提出的基于重用流行口令的用户行为，利用条件概率，进而使用口令重用模型评估非定向猜测下口令强度的技术路线是切实有效的。

第二节 相关工作

口令强度评估器能够为用户提供快速、准确的口令强度反馈，进而引导用户选择更加安全的口令。因此，相当多的网站在用户注册账户时提供口令强度评估服务，学术界和工业界也设计了各类口令强度评估器，旨在让口令强度评估更加准确有效。在工业界应用最为广泛的一类口令强度评估器，当属基于模式的口令强度评估器。这类口令强度评估器人为定义了一系列基于口令字符的规则（如口令中字母、数字、特殊字符等字符种类的数量），并利用这些规则来评价口令的安全性。比如，12306-PSM 和 MS-PSM 就是典型的基于模式的口令强度评估器，分别由 12306 和微软两大公司设计，其为上千万用户提供口令强度评估服务，代表了工业界所设计口令强度评估器的普遍水平。工业界所设计的最为先进的口令强度评估器之一，则是 Zxcvbn-PSM[43]。该口令强度评估器由 Dropbox 公司提出，使用了精心设计的启发式方法，并能够结合口令的语义信息对口令强度进行评价，并被部署于 Dropbox、BitWarden 等网站[27, 29, 44]，被学术界的有关研究认为是工业界设计的最准确、最有效的口令强度评估器[21, 24]。

相较于工业界，学术界设计的口令强度评估器，较少使用简单的模式规则和启发式方法，而更强调科学性和理论基础。以 PCFG-PSM[61]、Markov-PSM[46] 为代表的基于口令猜测模型的口令强度评估器，其从攻击者视角出发，将口令猜测模型对口令赋予的概率视为口令抗猜测性的表征，借此来评估口令强度。以 LPSE 为代表的口令强度评估器则基于口令与强口令的相似度，估计用户口令与强口令之间存在的差别，进而评估用户口令强度。以 CNN-PSM[62]、RNN-PSM[39] 为代表的基于深度学习的口令强度评估器，使用了人工智能领域提出的轻量级深度学习模型，利用数据驱动的方法，从口令数据集里提取口令语义信息，实现了更加准确且在一定程度上可解释的口令强度评估方法。

值得注意的是，上述口令强度评估器均用于评估非定向口令猜测攻击下的口令强度。尽管非定向口令猜测攻击并没有显式地利用口令重用行为，但仍有研究指出有必要在非定向口令强度评估中考虑口令重用行为[21, 22]。为了说明这一点，Wang 等人提出了 fuzzyPSM [23]，该口令强度评估器基于启发式方法对口令重用行为进行建模，并能够利用统计学模型从两个不同口令数据集之间提取、捕捉口令重用行为，最终基于口令重用行为评估口令在非定向猜测攻击下的强度。一系列后续研究认为，基于口令重用行为的 fuzzyPSM 是迄今为止学术界提出的最为准确有效的口令强度评估器[21, 24]。

FuzzyPSM 的成功说明，有必要在非定向口令强度评估中考虑口令重用行为。对口令重用行为进行建模的最有效方法之一是使用口令重用模型[22, 48]，然而在非定向口令强度评估中使用口令重用模型却困难重重。首先，口令重用模型需要用户的旧口令作为输入，而用户的旧口令是敏感的个人隐私信息，难以被服务提供商获取[59]；其次，口令重用模型的设计初衷是模拟或实施口令重用攻击，现有的基于口令重用模型的口令强度评估器（如 BERT-PSM[14]、PR-PSM[48]）均用于评估口令在口令重用攻击下的强度，难以在非定向口令强度评估中直接使用。此外，现有的基于口令重用模型的口令强度评估器，均需要利用口令重用模型生成包含大量口令的字典，进而计算口令被攻破所需的猜测数目[14, 48, 58]。这一过程需要较大算力成本和时间开销，为口令重用模型在非定向口令强度评估中的应用带来了更大挑战。

第三节　预备知识

2.3.1　口令数据集

表 2.1 本章所使用的真实口令数据集

数据集名称	服务类型	语言	泄露时间	口令总数	独立口令数
Tianya	社交网站	中文	2011 年 10 月	30,901,241	12,898,437
Sina	门户网站	中文	2011 年 12 月	19,383,163	3,748,140
Dodonew	电子商务	中文	2011 年 12 月	16,258,891	10,135,260
Zhenai	异性交友	中文	2011 年 10 月	5,260,229	3,521,764
Weibo	社交网站	中文	2011 年 12 月	4,730,662	2,828,618
Linkedin	求职应聘	英语	2012 年 6 月	54,656,615	34,334,121
Rockyou	社交网站	英语	2009 年 12 月	32,575,500	14,330,075
Yahoo	门户网站	英语	2012 年 7 月	5,626,485	3,439,492
Gmail	邮箱服务	英语	2014 年 9 月	4,929,086	3,119,299
Phpbb	技术论坛	英语	2009 年 1 月	255,421	184,389

本章所使用的真实口令数据集如表2.1所示。其中，每个数据集的名称均来自其网站域名或公司名称；“服务类型”则指数据集所对应的网站提供的服务类

型，如 LinkedIn 主要为用户提供求职、招聘的平台；语言则指数据集所对应网站的主要用户群体所使用的语言；“泄露时间”指口令数据集被攻击者公开在互联网的时间。可以看到，本章所选取的数据集里，中文和英文数据集各有 5 个，服务类型较为多样，且包含的口令数目庞大，具有较好的代表性。使用这些来源多样、种类丰富的大规模真实口令数据集，有助于更好探索我们所提出的 EditPSM 口令强度评估器的准确性和有效性。

需要特别注意的是，尽管上述口令数据集被攻击者公布于互联网，并且已经被研究者用于口令安全相关的学术研究，这些数据集本质上依然是用户隐私信息。出于学术伦理考虑，为了避免本章所使用的口令数据集对用户隐私造成进一步的伤害，本章在记录并分析实验结果时，仅公开相应的统计数据，不公开口令的具体信息（除极少数作为样例的口令样本）。此外，在进行本章的相关实验时，我们不保留口令之外的用户相关信息（如用户名、账户、电话号码等），避免口令被用于登录相关用户的账户。同时，我们不再进一步传播、分发这些口令数据集，防止攻击者获取这些口令并进行恶意攻击。

2.3.2 Pass2Edit 口令重用模型

在选取合适的口令重用模型，并将其应用于口令强度评估时，我们需要综合考虑这些模型的重用行为建模精确度、所占用的存储空间、使用的算力资源。一些具有代表性的口令重用模型所需要占用的存储空间如表2.2所示，可以看到其中 Wang 等人提出的 Pass2Edit[51] 体积占用较小。同时，Pass2Edit 是较为先进的口令重用模型，能够较为准确地对口令重用行为进行建模。因此，在本章我们基于 Pass2Edit 重用模型作为基础模型，来构建 EditPSM 口令强度评估器。对于其它口令重用模型的改造及其在口令强度评估中的应用，我们将在第三章进行介绍。

表 2.2 不同口令重用模型所占用的存储空间大小

Model	Targuess-II[22]	wang2023rfguess[13]	Pass2Path[47]	Pass2Edit[51]
Size	1.04G	121M	40.1M	8.62M

Pass2Edit 口令重用模型将口令重用视为一系列字符级编辑操作，这些编辑操作由用户选取，并应用于用户的原始口令，最终形成新的口令。字符级编辑操作本身作为一种原子操作，是逐一作用在用户口令上的，Pass2Edit 利用深度学习模型，将其用多步骤决策机制进行建模，最终精准捕捉用编辑操作进行描述的口令重用行为。具体来说，给定用户的旧口令和新口令，Pass2Edit 学习将旧口令修改为新口令所需的一系列字符级编辑操作。形式上，考虑一对用户口

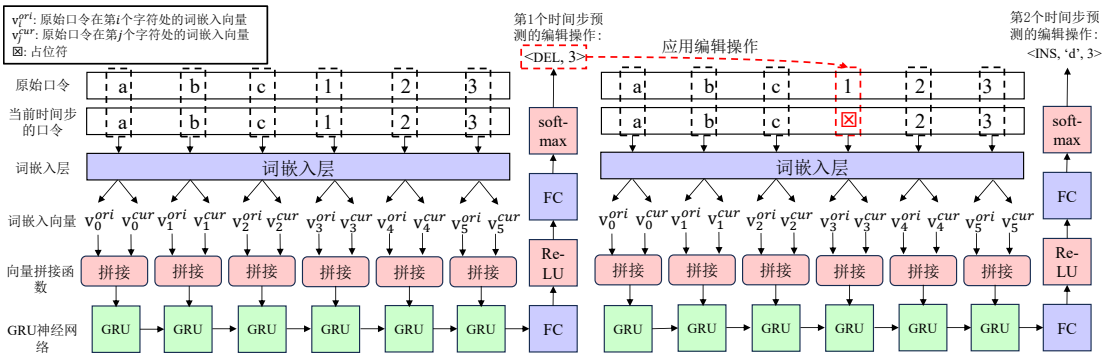


图 2.1 Pass2Edit 的模型架构

令 $\langle pw_1, pw_2 \rangle$，我们将其称为口令对，其中 pw_1 和 pw_2 分别为用户的旧口令和新口令，并将从 pw_1 转换为 pw_2 所需的编辑序列定义为 $\mathcal{T} = t_1, t_2, ..., t_k$，其中 t_i 为第 i 步所需的字符级编辑操作，其定义如下：

$$t_i \in \{(INS, p, c) | p \in \mathbb{N}, c \in \Sigma\} \cup \{(DEL, p) | p \in \mathbb{N}\} \cup \{EOS\}, \tag{2.1}$$

其中，p 和 c 分别表示要插入或删除的位置和字符，INS 和 DEL 代表插入和删除操作，Σ 表示字符集，$\mathbb{N}$ 表示自然数集。上述编辑操作和两个口令之间的编辑序列可以由动态规划方法得到，当我们得到多个可能的编辑序列时，我们将选取其中编辑操作总数最小的一个；当存在多个编辑操作总数最小的路径时，我们随机选取其中的一条路径。在生成口令猜测时，Pass2Edit 模型对于给定的用户口令，在每个时间步生成一个字符级编辑操作，并将该编辑操作应用于给定口令上，继续进行下一步的编辑操作生成。当我们将口令的最大长度限制为 30 个字符时，不难计算得到编辑操作的全部种类数目一共为 1531 类。这样一来，Pass2Edit 根据用户旧口令生成新口令的过程，本质上就是一个类别数为 1531 的多步骤分类决策问题。

如图2.1所示，Pass2Edit 以 GRU 神经网络[63] 为基础，构建了一个以口令作为输入，输出编辑操作的模型。对于一个时间步，Pass2Edit 首先通过词嵌入层获得原始口令和上一时间步口令中各个字符的词嵌入，然后再将原始口令、上一时间步口令各个字符的词嵌入进行对位拼接，共同传入三层 GRU 神经网络中。这种对位拼接的方式有助于模型学习过往时间步一系列编辑序列对口令的编辑效果，进而更好预测当前时间步所需要输出的编辑操作。随后，GRU 输出将会通过全连接层和 softmax 函数，得到编辑操作上的概率分布，进而得到当前时间步所需要使用的编辑操作。

2.3.3 评价指标

本质上来讲，非定向口令强度评估是对目标口令数据集里各个独立口令频率排名的估计[23]。换言之，在理想情况下，一个完全准确的、完美的理想口令强度评估器应当能够完全准确地重现目标口令数据集里面各个口令的概率：

$$M(pw) = p_{pw}, \forall pw \in \Gamma, \tag{2.2}$$

其中，$M(\cdot)$ 为理想口令强度评估器，p_{pw} 表示口令 pw 在目标口令数据集 Γ 中的概率。任何一个口令强度评估器都可以视为是对理想口令强度评估器的一种逼近。这样一来，在对非定向口令强度评估器的准确性进行评价时，我们只需要将该口令强度评估器的输出和完美口令强度评估器的输出进行比较即可。形式上，非定向口令强度评估的准确性由加权 Spearman 系数（$WSpearman$）进行评价：

$$WSpearman = \frac{\sum_{i=1}^{n}\left[w_i\left(x_i-\bar{x}\right)\left(y_i-\bar{y}\right)\right]}{\sqrt{\sum_{i=1}^{n}\left[w_i\left(x_i-\bar{x}\right)^2\right]\sum_{i=1}^{n}\left[w_i\left(y_i-\bar{y}\right)^2\right]}}, \tag{2.3}$$

其中，x_i 和 y_i 分别为待评价的口令强度评估器、理想口令强度评估器输出的结果列表 X 和 Y 中的第 i 个元素，w_i 则表示每个元素的权重，对应着口令数据集里第 i 个口令的频数（按频数降序排列）。$\bar{x}$ 和 $\bar{y}$ 分别代表 X、Y 的加权平均数。$WSpearman$ 的取值范围为 $[-1,1]$，数值越大则待评价的口令强度评估器的输出结果与理想口令强度评估器越相似，即待评价口令强度评估器的准确性越高。

考虑到 $WSpearman$ 是一个单一的值，为了体现口令强度评估器的准确性随口令排名（即口令流行度）的变化，我们进一步使用 $WSpearman$ 曲线[21] 来展现口令强度评估器的准确性。由于理想口令强度评估器只能对高于一定频率阈值的口令输出合理的概率，所以我们首先让待测评口令强度评估器对数据集里频率大于等于 10 的口令进行强度评估，提前得到整个数据集里的 X 和 Y 两个列表。然后，对于任意排名为 r 的口令，对应的 $WSpearman$ 值为：

$$WSpearman_r = WSpearman(X[:r], Y[:r]), \tag{2.4}$$

其中 $X[:r]$ 和 $Y[:r]$ 分别表示 X 和 Y 中的前 r 个元素。考虑到许多口令强度评估器会对不同的口令产生相同的输出，如果列表切片里的所有元素均完全相同，将会导致出现分母为零的情况。因此，我们规定，当 n 个口令 $[a_j^0, a_j^1, ..., a_j^n]$ 的频数相同时，我们将它们的排名计算为：

$$rank_j = a_j + \frac{\sum_{k=0}^{i} w_k}{i} \times \frac{i+1}{2}, \tag{2.5}$$

其中 w_k 是在带有相同频数的口令中排序为第 k 的口令的频数。

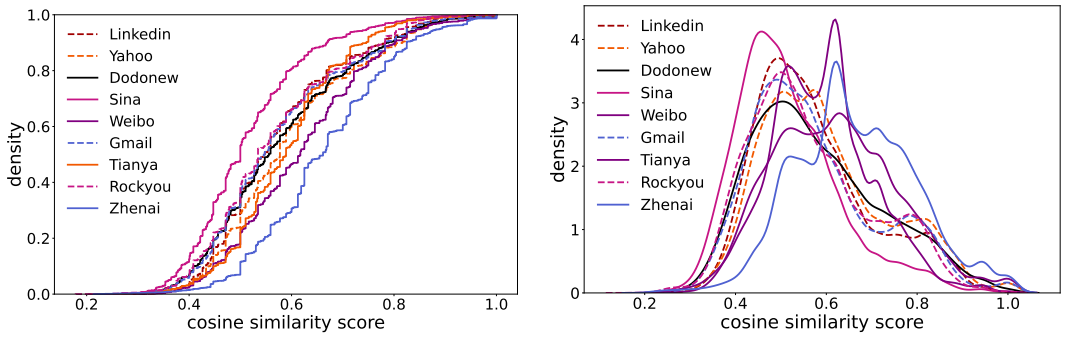
2.3.4 本章比较的现有非定向口令强度评估器

工业界设计的口令强度评估器。 12306-PSM[28] 和 MS-PSM[26, 42] 分别由 12306 和微软公司设计，用于在账户注册中评估口令强度。由于这两种口令强度评估器均基于口令字符种类进行强度评估，并且其被用于服务上千万用户，二者的准确性可以代表工业界口令强度评估器的普遍水平。此外，一系列研究表明，Zxcvbn-PSM[43] 是工业界设计的最为先进的口令强度评估器，受到了工业界和学术界的广泛认可[21, 24]。将这三个口令强度评估器和 EditPSM 进行比较，有助于体现 EditPSM 相对工业界普遍水平和先进水平的优势。

基于统计学模型的口令强度评估器。 PCFG-PSM[46] 基于 PCFG 口令概率模型，其采用攻击者的视角对口令强度进行评估，利用 PCFG 猜测模型输出的口令概率作为口令强度的表征。在 PCFG-PSM 的基础上，fuzzyPSM[23] 利用精心设计的启发式重用行为建模方案，让 PCFG 模型能够有效捕捉两个不同数据集之间存在的口令重用行为，进而在评价口令强度时能够考虑用户的口令重用行为。大量研究表明，fuzzyPSM 是迄今为止最为准确的非定向口令强度评估器[21, 24, 64]。将二者和 EditPSM 一起进行比较，可以展现 EditPSM 相对统计学模型的准确性提升。

CNN-PSM 和 RNN-PSM。 CNN-PSM[62] 基于卷积神经网络进行设计，能够捕捉口令的语义特征，针对用户口令给出字符级别的强度反馈和修改建议，具有较好的可解释性。RNN-PSM[39] 则基于循环神经网络，首先设计了基于深度学习的口令猜测模型，然后再根据该模型所输出的口令概率作为强度表征，准确性较高。由于我们所使用的基础口令重用模型 Pass2Edit[51] 同样利用了深度学习技术，将二者与我们所提出的方案进行比较，有助于展现我们方案所使用深度学习技术的有效性。

LPSE。 LPSE[65] 将口令强度视为用户口令与一个预定义的、强度较高的口令之间的相似度，相似度越高则口令强度越高。后续我们将根据 Pass2Edit 构建基于流行口令重用行为的 EditPSM 口令强度评估器，本质上是根据用户口令



(a) 累积分布 (b) 概率分布

图 2.2 余弦相似度分布。

和流行口令之间的重用“距离”来评估口令强度。LPSE 的设计理念与我们的 EditPSM 正好相反，所以将 LPSE 与我们的 EditPSM 进行对比，有助于揭示我们从用户行为出发构造口令强度评估器的技术路线是否有效。

上述部分口令强度评估器在实验中的有关设定细节，参见附录第一节。

第四节 基于流行口令的口令重用行为

为了探索用户是否重用流行口令，我们对口令数据集进行了一些统计。我们采用二阶余弦相似度来评价两个口令之间的相似度：

$$\mathrm{sim}(pw_A, pw_B) = \frac{\sum_{g\in\mathbb{G}} (\mathrm{count}(pw_A, g) * \mathrm{count}(pw_B, g))}{\sqrt{\sum_{g\in\mathbb{G}} \mathrm{count}^2(pw_A, g)} \sqrt{\sum_{g\in\mathbb{G}} \mathrm{count}^2(pw_B, g)}}, \tag{2.6}$$

其中，$\mathbb{G}$ 是口令 pw_A 和 pw_B 里面所有连续二元字符串的集合，$count(pw, g)$ 则表示 pw 中二元字符串 g 的数目。二阶余弦相似度的取值范围是 $[0, 1]$，越大代表两个口令越相似。然后，对于每个数据集，我们将其中频数最高的前 1000 个口令作为流行口令词典，然后随机选取数据集里面剩余的 10000 个口令与这 1000 个流行口令两两进行相似度计算。对于随机抽取的口令，我们将其与流行口令的相似度定义为与这个口令与所有 1000 个流行口令的余弦相似度最大值。

每个数据集里随机抽取的口令与流行口令的相似度分布如图2.2所示。可以看到，每个口令数据集里面的口令，有相当一部分与流行口令是高度相似的。从图2.2(b)中不难发现，每个口令数据集里，用户口令与流行口令的相似度概率分布曲线存在多个峰值，说明其分布并不是均匀的。我们认为，上述统计结果说明有相当大比例的用户重用流行口令，进而导致其口令与流行口令高度相似。

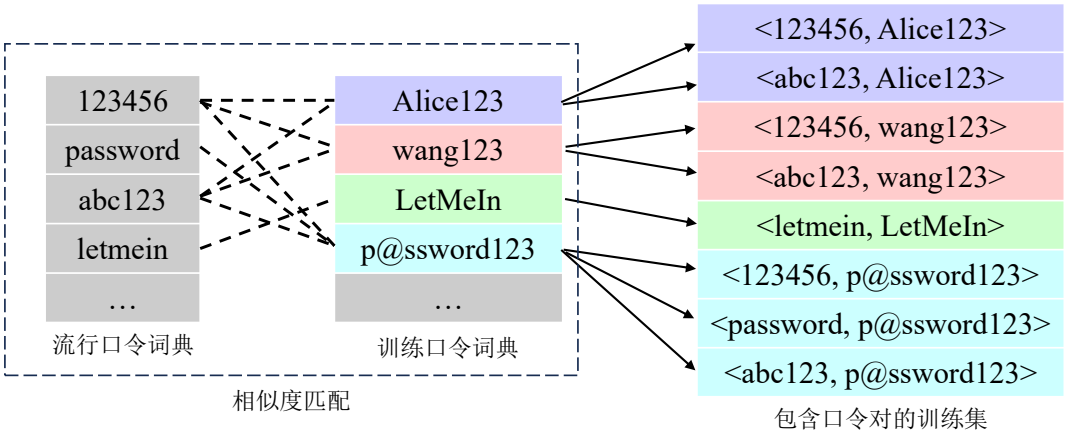


图 2.3 使用口令数据集构造口令重用模型的训练集。该训练集中的每个口令对均包含一个用户口令和一个与其高度相似的流行口令，口令重用模型可以学习用户重用流行口令的行为。

第五节 基于 Pass2Edit 的非定向口令强度评估器

在本章第四节中，我们发现大量用户重用流行口令（例如，将 *123456* 修改为 *123456!*）。基于这一发现，我们可以通过口令重用模型来建模这种基于流行口令的重用行为，进而利用口令重用模型构造非定向口令强度评估器。

首先，我们需要让 Pass2Edit 成功捕捉用户重用流行口令的重用行为。口令重用模型的训练数据是口令对，每个口令对包含用户的两个口令。随后，口令重用模型从中学习用户如何通过重用旧口令来选择新口令，进而建模用户的口令重用行为。为了让 Pass2Edit 学习用户重用流行口令的行为，我们利用口令数据集构造包含用户口令和流行口令的一系列口令对。如图2.3所示，给定一个流行口令词典和训练口令词典（即用于训练模型的口令数据集），对于训练口令词典的每个口令，我们将其与流行口令词典中的所有流行口令进行相似度计算。然后，选取其中相似度最高、且相似度分数大于阈值 *thresh* 的 *k* 个流行口令，不足 *k* 个则仅取相似度分数高于 *thresh* 的流行口令。随后，这 *k* 个流行口令将和用户口令构成 *k* 个口令对。上述流程结束之后，我们即可获得包含一系列口令对的训练集。在这个训练集里面，每个口令对都包含一个用户口令和另一个与之高度相似的流行口令。Pass2Edit 模型在这些口令对上训练之后，即可学习用户重用流行口令的行为。

随后，我们将训练好的 Pass2Edit 模型作为基础模型，构建 EditPSM 口令强度评估器。给定一个待评测的用户口令，EditPSM 将其与流行口令词典中的流行口令逐一进行相似度计算，选取其中相似度最高的流行口令。然后，EditPSM 提

取从流行口令变换得到该用户口令所需要的多步骤编辑序列，将编辑序列和每一个时间步编辑得到的中间结果输入到 Pass2Edit 模型中。这样一来，Pass2Edit 即可得到每一个编辑操作的概率。形式上，我们最终通过 EditPSM 得到的每一步编辑操作 $t = t_1, t_2, , EOS$ 的概率，最终累乘起来，就是用户口令 pw_1 由流行口令 pw_2 重用而来的条件概率：

$$\Pr(pw_1|pw_2) = \Pr(t_1|pw_1, pw_1) \times \Pr(t_2|pw_1, pw_i^{curr}) \times ... \times \Pr(EOS|pw_1, pw_n^{curr}), \tag{2.7}$$

其中，pw_i^{curr} 是将编辑操作 t_i 应用到 pw_{i-1}^{curr} 得到的当前时间步口令. 由于 pw_2 是流行口令, 条件概率 $\Pr(pw_1|pw_2)$ 越高，用户口令 pw_2 的安全性越低。这是因为这个条件概率本质上刻画了用户基于流行口令 pw_2 来创建 pw_1 的概率，条件概率越高，说明用户重用流行口令的方式越简单、越易于猜测，也就是越容易被非定向口令猜测攻击成功破解。

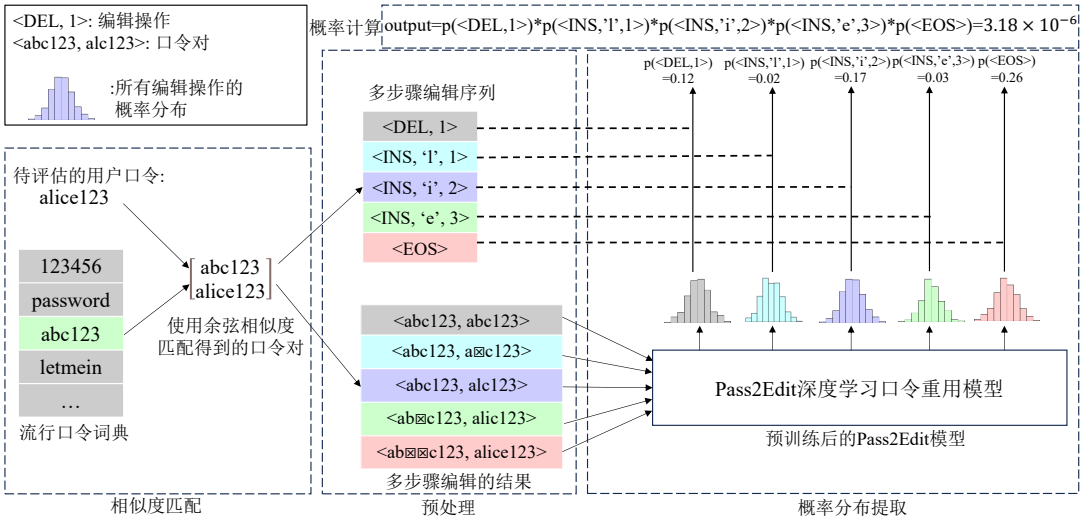


图 2.4 利用 Pass2Edit 模型构建 EditPSM 口令强度评估器。

第六节 实验设计与结果

2.6.1 实验场景设定

由于本章主要关注在线非定向口令猜测攻击中的口令强度评估，我们选取现有研究所广泛推荐的 10^4 次口令猜测作为在线非定向口令猜测的阈值。实施在线非定向口令猜测攻击的攻击者主要可以分为以下两类：

先验知识充分的攻击者。我们认为，能力较强、先验知识充分的攻击者，可以在

实施非定向猜测攻击之前提前获知目标网站、身份认证系统中用户的口令分布。这种获知了目标用户口令分布的攻击者，可以直接按照真实口令概率分布，按概率降序逐一尝试口令猜测，进而在攻击中取得理论上最高的口令破解率。这种强大的、先验知识充分的攻击者是完全可能存在的，因为现有案例表明，有许多网站和服务提供商（例如 Yahoo 和 Twitter[32, 66]）不止一次地泄露了其用户的口令文件，导致攻击者可以直接获取其用户群体的真实口令分布。为了评价 EditPSM 和其它现有口令强度评估器在防御先验知识充分的攻击者时的防御效果，我们采用 Wang 等人在 2023 年提出的评价方法：在每个测试口令数据集里面，直接让口令强度评估器输出所有口令的强度，然后计算 *WSpearman* 分数[21]。

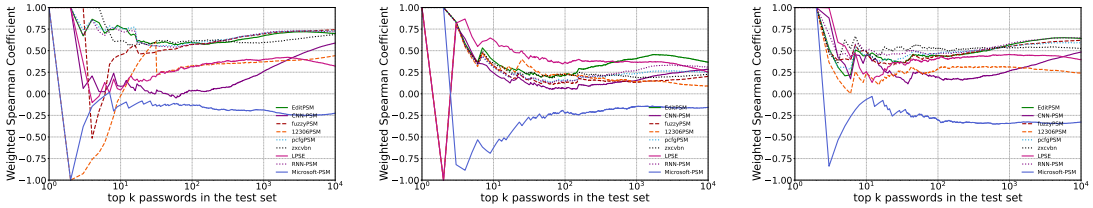
能力有限的攻击者。对于能力有限的攻击者来说，其无法得到完全准确的目标网站、身份认证系统中用户的口令分布[21]。对于这类攻击者来说，他们的最佳策略是在进行猜测攻击时，尽可能逼近目标网站、身份认证系统中的用户口令分布。为了模拟能力有限攻击者，我们采用 Wang 等人提出的模拟方案，即在测试集里随机选取 10^4 的口令进行测试，并计算各口令强度评估器在评价这些口令强度时的 *WSpearman* 值。

本章使用 10 个真实口令数据集，构建了 6 个不同的口令强度评估场景，以分别评价口令强度评估器在面对先验知识充分和能力有限两类攻击者时的准确性。如表2.3所示，我们在中英文两个语言上分别设计了 3 个测试场景。考虑到有 5 个口令强度评估器需要使用训练集，我们采用 Wang 等人提出的测试方法，分别采用规模大、性质良好、广为研究的 Tianya 和 Rockyou 数据集作为中英文两种语言上的训练数据集。

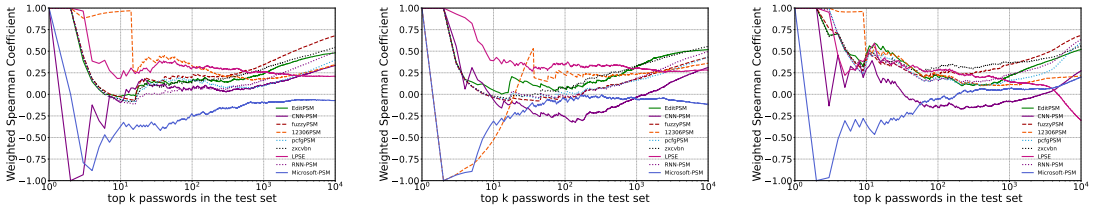
表 2.3 口令强度评估器使用的训练和测试数据集

场景编号 #	语言	训练数据集 A	训练集 B*	测试数据集
1	中文	Tianya	Zhenai	Weibo
2			Weibo	Sina
3			Weibo	Dodonew
4	英文	Rockyou	Phpbb	Linkedin
5			Gmail	Yahoo
6			Yahoo	Gmail

* 训练数据集 B 仅由 fuzzyPSM[23] 使用, 其需要使用两个不同的数据集来捕捉数据集之间存在的用户重用行为。

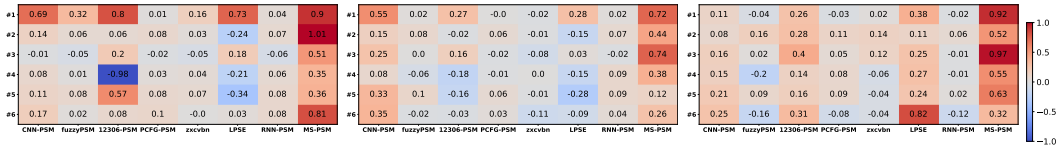


(a) #1, Tianya→Weibo (b) #2, Tianya→Sina (c) #3, Tianya→Dodonew



(d) #4, Rockyou→Gmail (e) #5, Rockyou→Linkedin (f) #6, Rockyou→Yahoo

图 2.5 抵御先验知识充分攻击者的实验结果。



(a) Rank= 10. (b) Rank= 10^2. (c) Rank= 10^4.

图 2.6 在不同标志性排名位置的相对 *WSpearman* 分数热力图。

2.6.2 实验结果

抵御非定向口令猜测攻击的准确性

总体而言，与学术界和工业界此前提出的口令强度评估器相比，EditPSM 的准确度更高。与工业界设计的口令强度评估器进行比较时，本章提出的 EditPSM 显著优于 12306-PSM[28] 和 MS-PSM[26]。具体而言，如图 2.6和 2.8所示，相对 12306-PSM 和 MS-PSM，在抵御先验知识充分和能力有限的两种攻击场景下，EditPSM 在各个标志性口令排名位置的 *WSpearman* 值分别平均提升了 0.222 和 0.6833。同时，如图 2.6 所示，EditPSM 与 zxcvbn [43]（工业界准确性最佳的口令强度评估器）性能相当，并且在针对极不安全口令（即排名在 10 和 10^2 之间的口令，参见图2.8）时，平均提升了 0.045。

对于基于相似性度量的口令强度评估器，可以看出 EditPSM 的表现显著优于 LPSE[65]：在所有 3 个标志性口令排名位置上，EditPSM 的 *WSpearman* 值平均提升了 0.175。而相较于基于统计学模型的 PCFG-PSM[61]，如图 2.5 和 2.6 所示，EditPSM 在抵御先验知识充分的攻击者时，在各个标志性口令排名上实现了 0.047 的提升；而在抵御能力有限的攻击者时，EditPSM 的 *WSpearman* 值也

实现了 0.0029 的平均提升。具体而言，如图 2.8 所示，EditPSM 在 $rank = 10$ 处的平均 *WSpearman* 比 PCFG-PSM 高出 0.137，这表明 EditPSM 所采用的深度学习技术在一定程度上缓解了 PCFG-PSM 等统计学模型面临的过拟合问题。

此外，我们还将 EditPSM 与两个基于深度学习的口令强度评估器（即 CNN-PSM[62] 和 RNN-PSM[39]）进行比较。如图 2.8 和 2.6 所示，在基于深度学习的口令强度评估器中，EditPSM 表现最佳。具体来说，EditPSM 在抵御两类攻击者时，准确性大幅优于 CNN-PSM[62]，在各个标志性口令排名位置上平均提升了 0.18。同时，EditPSM 比 RNN-PSM[39] 具有更高的准确性，实现了 0.032 的 *WSpearman* 提升。上述结果表明，本章提出的 EditPSM 在使用深度学习技术评估口令强度时，比现有的深度学习口令强度评估方案更加准确。

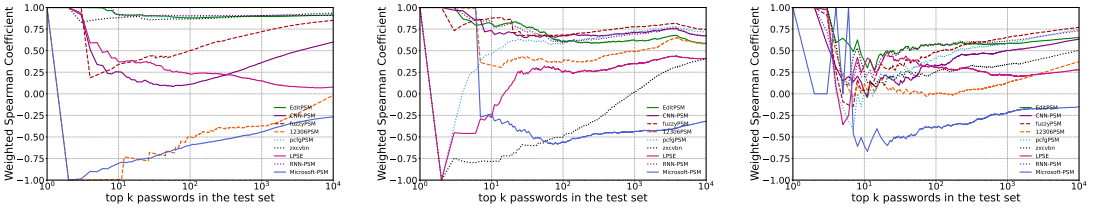
迄今为止，根据 Wang 等人[21] 的研究，fuzzyPSM 是目前为止最为准确的非定向口令强度评估器。与 fuzzyPSM 相比，本章提出的 EditPSM 同样具有一定的优势。如图2.5 和 2.6 所示，在抵御先验知识充分的攻击者时，本章提出的 EditPSM 实现了比 fuzzyPSM[23] 略高的准确性，在所有的标志性口令排名上的 *WSpearman* 平均提升了 0.0254。我们认为，这一结果可以归功于本章所提出的新技术路线：EditPSM 可以捕捉更细粒度的口令层面重用行为，而 FuzzyPSM 本质上学习的是数据集层面的宏观口令重用行为，其粒度较粗，难以精准刻画用户复杂的口令重用行为。

模型性能评估

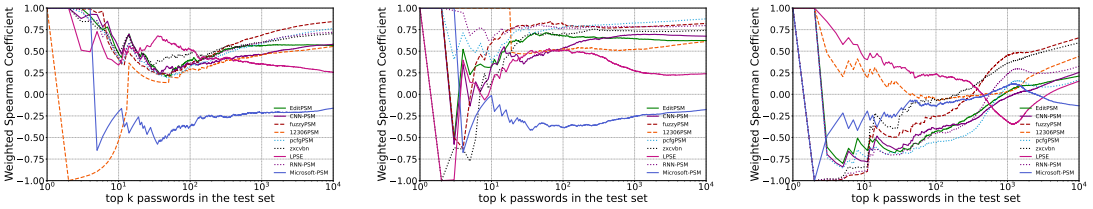
考虑到深度学习模型相较统计学模型和启发式方法，需要更多的算力资源和存储空间，我们将 EditPSM 的运行速度和占用体积和现有的基于深度学习的口令强度评估器进行了比较。如表2.4所示，我们在单张 NVIDIA GeForce RTX 3090 GPU 上分别运行了 EditPSM 和 CNN-PSM、RNN-PSM，记录了其推理速度和模型所需要占用的硬盘空间。和现有的深度学习口令强度评估器相比，我们的 EditPSM 所需占用的存储空间最小，同时对每个口令所需要的平均推理时间达到了毫秒级。基于以上实验结果，可以认为 EditPSM 所需的存储空间和推理速度，对于绝大多数用户而言是能接受的，初步达到了在客户端常见设备上部署的性能要求。

第七节　本章小结

在本章里，我们首次解决了“如何使用口令重用模型评估非定向猜测攻击下的口令强度”这一问题。我们通过观察用户的口令重用行为，从大规模的真

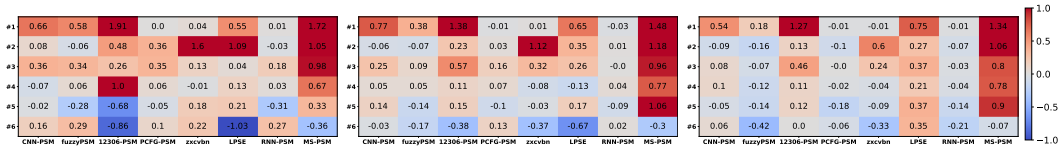


(a) #1, Tianya→Weibo (b) #2, Tianya→Sina (c) #3, Tianya→Dodonew



(d) #4, Rockyou→Gmail (e) #5, Rockyou→Linkedin (f) #6, Rockyou→Yahoo

图 2.7 抵御能力有限攻击者的实验结果。



(a) Rank= 10. (b) Rank= 10^2. (c) Rank= 10^4.

图 2.8 在不同标志性排名位置的相对 *WSpearman* 分数热力图。

表 2.4 基于深度学习的口令强度评估器的体积占用和推理速度

口令强度评估器	单个口令的评估所需时长	模型存储所需体积
EditPSM	2.71ms	8.63M
CNN-PSM[62]	4.74ms	88.4M
RNN-PSM[39]	1.72ms	8.77M

实口令数据集里发现用户不仅重用旧口令，还会重用流行口令，进而拓宽了口令重用行为的边界。基于这一关键发现，我们进一步提出用户使用弱口令的行为本质上可以由重用流行口令的重用行为进行刻画。随后，我们利用 Pass2Edit 口令重用模型，构建了 EditPSM 口令强度评估器，其能够有效捕捉用户重用流行口令的行为，进而评估口令在非定向猜测攻击下的强度。

通过大规模的实验，我们将本章提出的 EditPSM 与现有的一系列代表性口令强度评估器，在非定向口令强度评估场景下进行了系统性的对比。实验结果表明，EditPSM 在现有的口令强度评估器中，准确性已经达到了先进水平，其模型体积、推理速度也已经初步达到了在客户端常见设备上部署的性能要求。该技术路线首次成功将口令重用模型应用于非定向口令强度评估中，为多功能口

令强度评估器的设计奠定了基础，相关内容将在第三章中进行介绍。

第三章　多功能口令强度评估器的设计

口令猜测攻击存在多个不同种类。如何设计能够在不同猜测攻击场景下评估用户口令强度的多功能口令强度评估器，是口令强度评估器需要面临的现实问题。本章主要介绍如何从用户两类口令重用行为的复杂度差异出发，通过持续学习技术，系统性地对现有的口令重用模型进行改造，构建能够同时抵御非定向口令猜测攻击、口令重用攻击的多功能口令强度评估器。

第一节　研究背景

口令猜测攻击存在多个不同种类，每一种攻击都会利用至少一种脆弱的用户口令行为。例如，非定向口令猜测攻击利用了用户倾向于使用弱口令的行为（如使用”123456” 或”password” 这类弱口令），口令重用攻击则利用了用户的口令重用行为（如基于旧口令”nankai520” 来创建新口令”Nankai520!”）。这些不同种类的口令猜测攻击各自能够捕捉的脆弱口令行为存在差别，所能够破解的口令也存在差异。例如，非定向口令猜测攻击擅长破解口令结构相对简单、较为流行的口令，却经常对结构相对复杂、长度较大、结合了个人信息的用户口令束手无策；相反，口令重用攻击能够基于用户的旧口令，通过构造旧口令的种种变体来进行口令猜测，往往能够破解结构上相对复杂，但与旧口令存在较高相似度的口令。例如，”yifeizhang2026nankai123” 属于长口令，结构相对复杂，仅用非定向口令猜测攻击难以破解；但如果攻击者获取了该用户的旧口令”yifeizhang2026nankai”，就可以轻而易举地使用口令重用攻击将其快速破解。所以，现实世界的攻击者经常同时使用多种口令猜测攻击，以利用不同种类的脆弱口令行为，以尽可能提高口令的破解率。

然而，通过口令强度评估器，来抵御这类同时针对不同脆弱口令行为、使用不同种类口令猜测攻击的联合攻击方式，则是困难重重。迄今为止，学术界和工业界尚未提出能够有效地同时建模不同口令猜测攻击下口令强度的评估方案。这主要是因为，对于口令强度评估而言，如果想要准确评估口令在一种口令猜测攻击之下的口令强度，必须准确捕捉这一口令猜测攻击所利用的脆弱口令行为。但是，当我们考虑多种不同的口令猜测攻击时，则需要同时精准建模不同种类的用户脆弱口令行为，这本身是一件非常困难的事情。一方面，据我们所知，目前来说仅有一款能够同时捕捉不同脆弱口令行为的模型，即 TarGuess-III∼IV 系列定向口令猜测模型，但这两个模型占用的存储空间过大（2G 以上），不利

于被改造为口令强度评估器；另一方面，如果使用多个模型来分别建模不同脆弱口令行为，则会带来额外的计算和存储开销。如何使用单个模型同时建模多种不同的脆弱口令行为，并构建多功能口令强度评估器，是同时抵御多种不同的口令猜测攻击所必须要解决的问题。

3.1.1 研究动机

我们在本文的第二章中提出，利用口令重用模型捕捉基于流行口令的重用行为，可以让口令重用模型评价非定向猜测攻击下的口令强度。由于口令重用模型本身也可以捕捉基于旧口令的口令重用行为，我们同样可以利用口令重用模型评价口令重用攻击下的口令强度。这样一来，利用口令重用模型就有可能同时评价非定向口令猜测攻击和口令重用攻击下的用户口令强度，进而成功构造一个多功能口令强度评估器。上述构想看似为构建多功能口令强度评估器指明了道路，实则存在一系列有待回答的关键问题：

- **用户的两种口令重用行为存在怎样的特点**？在第二章中，我们提出用户不仅重用旧口令，而且还会重用流行口令；用户选用弱口令的行为，可以被视为重用流行口令的行为。现在，由于我们需要进一步用口令重用模型同时建模两类口令重用行为，我们在拓宽口令重用行为边界的同时，需要解决的新问题是：重用旧口令、重用流行口令这两种不同的口令重用行为，各自存在怎样的特点、二者是否存在差异、有怎样的差异。准确回答上述问题，将为构建多功能口令强度评估器提供基本理论前提。
- **如何利用持续学习技术，让口令重用模型同时捕捉两种口令重用行为**？使用口令重用模型同时捕捉基于旧口令和流行口令的两种不同口令重用行为，需要使用持续学习来让同一个模型学会处理两种不同任务。然而，将持续学习技术直接应用在口令重用模型、口令强度评估任务上，存在诸多需要敲定的技术细节。例如，持续学习的训练过程存在顺序的先后之分，怎样在多功能口令强度评估的场景下，合理安排两种口令重用行为在持续学习中的训练顺序？持续学习本身不是多个任务训练过程的简单重复，而是需要将一个任务上学到的知识迁移到另一个任务上，那么在训练口令重用模型时，应当让模型“迁移”哪种口令重用行为上的哪些知识？解决上述问题，是将持续学习技术引入口令强度评估领域、构建多功能口令强度评估器的必要条件。
- **学术界提出了一系列口令重用模型，如何将他们系统性地改造为多功能口**

令强度评估器？近年来，学术界已经提出了一系列先进的口令重用模型，能够较为精准地建模口令重用行为。这些模型在结构、机理等各个方面存在区别，是否存在一种系统性的改造方案，能够适应性地根据这些口令重用模型各自的特点，来将这些口令重用模型改造为多功能口令强度评估器？研究并回答这一问题，将为多功能口令强度评估器的构建方案提供更高的可扩展性，不再局限于特定的模型架构、种类，以便于将未来提出的更先进口令重用模型改造为多功能口令强度评估器，实现更精准的口令强度评估。

3.1.2 本章贡献

围绕“如何构建多功能口令强度评估器”，本章的主要贡献如下：

- **基于统计学的观察**。本章基于大规模口令数据集，对两种口令重用行为的特点和区别进行了基于统计学的分析和观察，得出了三条关键结论：用户存在重用流行口令的行为，用户重用流行口令的行为比重用旧口令的行为复杂度更低，用户使用弱口令的行为本质上是一种口令重用行为。这三条结论为后续的持续学习打下了理论基础，为多功能口令强度评估器的设计提供了必要前提。
- **系统性设计原则与方案**。本章通过基于旧口令、流行口令的两种口令重用行为在复杂度、训练数据稀疏性等方面的差别，提出了使用持续学习训练口令重用模型的一系列原则，敲定了训练先后顺序、模型结构冻结等持续学习中的关键技术细节。得益于这些原则，本章首次利用持续学习技术，让口令重用模型同时准确建模基于旧口令、流行口令的两种口令重用行为，进而将口令重用模型改造为多功能口令强度评估器。此外，本章所提出的系统性设计方案，不依赖于口令重用模型的具体种类、结构、技术路线，能够有效地将现有的各类口令重用模型改造为准确性良好的多功能口令强度评估器，并可以兼容未来更先进的口令重用模型。
- **大规模实验**。本章通过持续学习技术，成功将现有的一系列口令重用模型（KNNGuess、PointerGuess、Pass2Edit、PassBERT、Pass2Path）改造为了多功能口令强度评估器。通过在 12 个大规模真实口令数据集上实施非定向猜测攻击、口令重用攻击两种口令猜测攻击实验，本章将改造得到的五款多功能口令强度评估器与现有的 11 款非定向口令强度评估器、重用口令强度评估器进行了比较。实验结果表明，本文提出的多功能口令强度评估

器设计方案，其成功改造口令重用模型所得到的五款口令强度评估器，能够在非定向猜测攻击、口令重用攻击两种攻击场景下，同时达到现有口令强度评估器的先进水平。

第二节 相关工作

我们将在本节介绍与多功能口令强度评估器相关的前人工作。其中，有关非定向口令强度评估器、非定向口令猜测攻击下口令强度评估的相关工作，本文已在第二章中进行介绍，此处不再赘述。

3.2.1 口令重用模型

口令重用模型以一个给定口令（如用户的旧口令）作为输入，建模用户根据该给定口令，选择新口令的重用行为。目前学术界已发表的有关工作中，所有口令重用模型均以单个给定口令作为输入，因此本文暂不考虑基于多个给定口令的口令重用模型。形式上，口令重用模型是一种概率模型，其建模的是用户基于给定口令 pw_{given}，选用新口令 pw_{new} 的条件概率 $\Pr(pw_{new}|pw_{given})$。不同种类的口令重用模型的主要区别，就在于其各自建模这一条件概率的方式上。

在 2014 年，Das 等人提出了首个口令重用模型。Das 等人首先通过用户调研、数据集统计等方式，收集了一系列口令重用行为的样例，并总结得到了一系列用于描述口令重用行为的启发式口令重用规则。随后，他们利用这些规则，设计了口令重用算法，其能够根据单个给定的用户口令，输出一系列修改后的口令，以模拟用户的口令重用行为并实施口令重用攻击。尽管 Das 等人提出的口令重用模型采用了根对重用规则进行概率排序等统计学方法，但他们对口令重用行为的建模依然是启发式的，无法捕捉用户口令中的语义、结构等关键信息，大大限制了其建模口令重用行为的精准性。

在 2016 年，Wang 等人基于 PCFG 统计学模型，提出了首个完全基于概率的口令重用模型 TarGuess-II。该模型采用了数据驱动的方案，在预定义的启发式重用行为基础上，建模了用户旧口令的语义，并考虑了旧口令本身对用户重用行为的影响。此外，TarGuess-II 显式建模了口令重用行为中口令的条件概率 $\Pr(pw_{new}|pw_{given})$，能够更精准、更高效地按照条件概率降序输出给定口令的不同变体，表达能力和口令重用攻击的成功率显著超越了 Das 等人在 2014 年提出的口令重用模型。

在 2019 年，Pal 等人首次基于深度学习模型提出了 Pass2Path 口令重用模型。Pass2Path 模型对口令条件概率的建模方式是，将口令重用行为视为用户对旧口

令的一系列编辑操作。模型只需要根据给定的用户旧口令，按照概率降序预测一系列编辑操作，即可得到旧口令的变体及其条件概率。形式上，Pass2Path 对口令条件概率的建模可以用编辑操作的累乘条件概率来描述：

$$\Pr(pw_{new}|pw_{given}) = \prod_{i=1}^{n} \Pr(\tau_i|\tau_0, ..., \tau_{i-1}, pw_{given}), \tag{3.1}$$

其中，$\tau_1 \sim \tau_n$ 为从 pw_{given} 编辑得到 pw_{new} 所需的一系列编辑操作，τ_0 为序列起始符。Pal 等人在上述理论的基础上，利用 Seq2Seq（Sequence-to-Sequence）模型在口令、编辑操作两类不同序列之间建立了桥梁，建模根据旧口令 pw_{given} 选择编辑操作序列 $\tau_1 \sim \tau_n$ 的过程。实验表明，相较于基于统计学模型的 TarGuess-II，使用深度学习技术的 Pass2Path 不再依赖启发式口令重用规则，能够更为准确地建模口令重用行为。

Wang 等人在 2023 年更进一步地将深度学习技术引入到口令重用模型中。他们发现，Pass2Path 模型虽然利用深度学习模型的泛化性强、表征学习能力强等一系列优势，但其采用的 Seq2Seq 模型难以有效捕捉编辑操作对用户口令的编辑效果，对编辑操作本身的语义感知能力有限。为了解决这一问题，Wang 等人在利用编辑操作建模口令条件概率的基础上，提出 Pass2Edit 口令重用模型。该模型将编辑序列的预测视为多步骤决策机制，每一个步骤的编辑操作都会相应对给定的用户口令产生编辑效果。具体来说，这种多步骤决策机制下的口令条件概率可以用如下表达式描述：

$$\Pr(pw_{new}|pw_{given}) = \prod_{i=1}^{n} \Pr(\tau_i|pw_{given}, pw_{i-1}), \tag{3.2}$$

其中，$\tau_1 \sim \tau_n$ 为从 pw_{given} 编辑得到 pw_{new} 所需的所有编辑操作，pw_{i-1} 为经过 $\tau_1 \sim \tau_{i-1}$ 编辑操作之后得到的当前时间步口令。这样一来，模型结合口令语义信息，可以精准捕捉每一步编辑操作对口令所产生的影响，更准确地建模用编辑操作描述的口令重用行为。

同样在 2023 年，Xu 等人利用 BERT 深度学习模型来改进对口令表征的学习，并设计 PassBERT 口令重用模型。该模型首先利用掩码语言建模技术（Masked Language Modeling）在口令数据集上对 BERT 模型进行预训练，然后利用 BERT 对语言中各个字符进行标签分配的能力，设计了一系列编辑标签来对口令条件概率进行建模。形式上，PassBERT 所建模的口令条件概率可以被描述为：

$$\Pr(pw_{new}|pw_{given}) = \prod_{c_i \in pw_{given}} \Pr(op_i|c_i, pw_{given}), \tag{3.3}$$

其中，c_i 是给定口令 pw_{given} 中的第 i 个字符，op_i 是对字符 c_i 进行的编辑操作，以标签形式呈现。实验结果表明，PassBERT 在口令重用攻击中实现了比 Pass2Path 更高的攻破率，凸显了其表征学习方案和条件概率建模方式的有效性。

2024 年，Xiu 等人成功突破了“利用编辑操作建模条件概率、描述重用行为”这一范数，并提出 PointerGuess 口令重用模型，不再使用编辑距离间接表示口令的重用方式，而是直接从字符层面对口令重用行为进行描述。PointerGuess 对口令条件概率的建模方式可以描述如下：

$$\Pr(pw_{new}|pw_{given}) = \prod_{i=1}^{|pw_{new}|} \Pr(c_i|c_0, ..., c_{i-1}, pw_{given}) \tag{3.4}$$

其中，c_i 为口令 pw_{new} 中的第 i 字符，$|pw_{new}|$ 为 pw_{new} 中所包含的字符总数。为建模上述条件概率，PointerGuess 使用指针机制，通过学习用户如何选择“保留”旧口令中的字符、直接选用全新字符，来对新口令的逐字符概率进行计算，最终得到整个新口令的条件概率。

Li 和 Wang 在 2026 年提出了基于检索增强生成的口令重用模型 KNNGuess[58]。该模型以 Transformer 模型 [67] 作为基础模型，利用训练数据构建检索数据库，在口令重用攻击中，直接利用训练数据中与用户旧口令相似的口令或口令片段，来为基础模型提供指导，以实现更高的破解率。KNNGuess 所建模的口令条件概率可以表述为：

$$\begin{aligned}\Pr(pw_{new}|pw_{given}) = \prod_{i=1}^{|pw_{new}|} (&\Pr_{Basic}(c_i|c_0, ..., c_{i-1}, pw_{given}) \cdot \lambda_{Basic} \\ &+ \Pr_{KNN}(c_i|c_0, ..., c_{i-1}, pw_{given}) \cdot \lambda_{KNN} \\ &+ \Pr_{Local}(c_i|c_j, ..., c_{i-1}, pw_{given}) \cdot \lambda_{Local})\end{aligned} \tag{3.5}$$

其中，$\Pr_{Basic}$ 是 Transformer 模型直接根据给定口令和当前已生成的新口令字符所输出的概率分布；$\Pr_{KNN}$ 是根据 Transformer 模型解码器根据给定口令、已生成的新口令字符所输出的高维表征，在数据库中检索训练数据并得到 k 个结果之后，根据这些检索结果得到的概率分布；$\Pr_{Local}$ 则是根据当前已生成的新口令字符中的片段，在数据库中进行检索得到的概率分布。将上述概率分布加权

求和之后，即可得到新口令在给定口令下的条件概率。

3.2.2 口令重用攻击下的口令强度评估

目前，口令重用攻击下的口令强度评估方案主要采取两种技术路线。其中一条技术路线，以 Vec-PPSM 作为代表，将新旧口令之间的相似度作为口令在重用攻击下的强度表征，相似度越高，则其抗猜测性越弱，越容易被破解。这一技术路线的主要缺陷是，简单的相似度指标（如词嵌入相似度、编辑距离等）表达能力有限，难以有效捕捉复杂的口令重用行为。例如，”!zyf2022!@#123” 和”2022zyf123” 在文本层面的相似度较高，通过”!zyf2022!@#123” 破解”2022zyf123” 的可能性很大，但仅通过”2022zyf123” 成功破解”!zyf2022!@#123” 则高度困难。而各类相似度指标仅从文本的相似度出发，无法捕捉口令重用行为所存在的这种方向性，进而可能会错误估计口令在重用攻击下的抗猜测性。

另外一种在口令重用攻击下的进行强度评估的技术路线是，通过口令重用模型模拟重用攻击，观察用户口令是否可以被成功破解（如 BERT-PSM 生成 1000 个口令猜测模拟口令重用攻击）。不幸地是，这种技术路线在准确性和计算复杂度上均存在缺陷。从准确性的角度来说，单个口令重用模型所能破解的口令是相对有限的，很难覆盖所有容易被口令重用攻击破解的口令。例如，当旧口令”caesar” 被泄漏时，BERT-PSM 根据该口令生成的 1000 个口令猜测并不包括用户的新口令”19caesar”，进而将该口令评价为安全口令，然而 TarGuess-II 可以在不到 100 次猜测内将其快速破解。从计算复杂度的角度来讲，利用口令重用模型实施口令重用攻击需要较长时间，且对算力资源有较高要求，为在客户端部署口令强度评估器带来了实际困难。例如，评价用户单个口令的强度时，PR-PSM 生成 1000 个猜测并评价口令强度需要 1.5 秒时间，并且需要 GPU 作为算力支持。

第三节 预备知识

3.3.1 口令数据集

本章在实验中所使用的大规模真实口令数据集如表3.1所示。我们在清洗数据集时，保留长度小于 30 且仅包含 ascii 可打印字符的口令；当数据集需要通过邮箱进行口令配对时，清理掉不附带邮箱信息的口令。上述数据清洗方式与学术界现有研究保持了一致。表中的“无效口令”指在数据清洗过程中被排除的口令，“是否包含邮箱”则表示数据集里是否包含了用户的邮箱信息，如果包含

表 3.1 本章所使用的真实口令数据集

数据集名称	语言	服务类型	总口令数目	泄露时间	是否包含邮箱	无效口令	无效占比%
Dodonew	中文	电子商务	16,282,286	Dec. 2011	是	256,016	1.57%
Tianya		社交论坛	30,816,592	Dec. 2011	是	9,062	0.03%
CSDN		技术论坛	6,428,410	Dec. 2011	是	3,164	0.05%
Taobao		电子商务	15,072,418	Feb. 2016	是	1,265	0.01%
126		邮箱服务	88,136,038	Dec. 2011	是	14,995	0.23%
Weibo		社交媒体	4,730,662	Dec. 2011	否	458	<0.01%
000webhost	英文	网站管理	15,299,907	Oct. 2015	是	116,462	0.76%
LinkedIn		招聘信息	54,656,615	Jan. 2012	是	122,051	0.22%
Twitter		社交媒体	25,575,929	May. 2016	是	287,551	1.12%
MathWay		教育培训	16,051,087	Jan. 2020	是	209,726	1.31%
Phpbb		技术论坛	255,421	Jan. 2009	否	10	<0.01%
Rockyou		社交媒体	32,575,500	Jul. 2012	否	9,864	<0.01%

则可以用于进行口令重用攻击下口令强度评估的相关实验。

本章的实验涉及口令重用攻击下的口令强度评估，需要在不同数据集之间，对同一用户在不同账户里使用的口令进行匹配。这一过程需要使用用户的邮箱信息，为避免可能的额外用户隐私泄露风险，我们采取一系列措施防范用户口令和个人信息的进一步泄露。首先，我们在获取同一用户在不同账户下的口令之后，立即删除用户邮箱、账号、用户名等一切其它隐私信息，避免相关信息被恶意获取。同时，我们在本文里仅记录研究得到的统计学结果，以及作为样例的个别用户口令文本，不记录用户口令的具体信息，也不披露相应的邮箱等个人信息。

3.3.2 口令重用数据集的构造

为了模拟现实世界中的口令重用攻防场景，我们利用十个包含用户邮箱的数据集，并设置了八种攻击场景（见表 3.2）。在这些场景中，攻击者使用 PointerGuess [48]、Pass2Path [47]、Pass2Edit[51]、PassBERT[14] 和 TarGuess-II [22] 进行口令重用攻击。同时，重用口令强度评估器将在这些攻击场景中对用户的口令强度进行评估。

首先，由于口令本身受到用户语言的较大影响[68]，我们将数据集划分为中文和英文两类。接下来，我们分别介绍八种攻击场景的设置和相关考量。场景 #1 模拟了攻击者获取用户在某服务的账户口令，并尝试攻击另一类似服务中账户口令的情形：Dodonew 和 Taobao 均为电子商务网站。场景 #2、#4、#6、#7 则模拟了攻击者从口令政策较弱（即对口令强度要求不高）的网站获取用户旧口令，并攻击口令政策较强（即对口令强度要求较高）网站中账户口令的情形。例如，000webhost 对用户口令的强度要求比 Twitter 更强。场景 3、5、8 模拟了攻击者在获取强关联口令后，尝试破解创建策略较弱账户的情况。由于攻击者可能在口令重用攻击中混入流行口令，以提升攻击成功率，我们参照现有研

表 3.2 口令重用攻击和防御场景的构造

编号	语言	训练集 (评估)* †	数目	迁移数据集 ‡	训练集 (攻击)†	数目	测试集 *	数目
#1	中文	126-Dodonew	49032	Tianya	Tianya-Dodonew	445514	Tianya-Taobao	455847
#2		Tianya-Dodonew	445514		126-Dodonew	49032	126-CSDN	57213
#3		Tianya-Dodonew	445514		CSDN-Dodonew	160199	CSDN-126	57213
#4		CSDN-Dodonew	160199		Tianya-Dodonew	445514	Tianya-CSDN	552263
#5	英语	LinkedIn-Twitter	300370	Rockyou	000webhost-Twitter	146945	000webhost-LinkedIn	214346
#6		000webhost-Twitter	146945		Twitter-LinkedIn	300370	Twitter-000webhost	295186
#7		LinkedIn-Mathway	111637		LinkedIn-Twitter	111637	LinkedIn-000webhost	214346
#8		Twitter-LinkedIn	300370		LinkedIn-Twitter	300370	LinkedIn-MathWay	111637

* 这类数据集由邮箱配对而来的姊妹口令对组成。
† 训练集（评估）用于训练重用口令强度评估器；训练集（攻击）用于训练攻击者的口令重用模型。
‡ 迁移数据集用于 VersaPSE 改造的各重用口令强度评估器在持续学习中使用。

究[21, 24] 的设定，使用每个测试集中前 1000 个最流行的口令作为攻击者的流行口令字典。我们利用上述攻击的结果来确定哪些口令在重用攻击下的安全性。具体来说，我们采用现有工作的常用设定[14, 22, 47, 48, 51]，以 1000 次猜测作为口令重用攻击的阈值：如果一个口令未出现在流行口令字典中 [21, 24]，且 TarGuess-II[22]、PointerGuess[48]、Pass2Path [47] 等口令重用模型均无法在 1000 次猜测内成功破解它，则该口令被视为是安全的。反之，则被视为不安全。

3.3.3 持续学习技术

持续学习技术[69, 70] 是在人工智能、深度学习领域常用的模型训练方法，其基本内容是让模型在多个任务上，按照一定先后顺序进行训练，进而让模型能够同时学会多个任务的处理。持续学习的有效性在于，合适地选择一系列任务的训练先后顺序，最小化训练过程中“遗忘”先前任务的风险，同时让模型能够利用不同任务中学习到的多种先验知识，更好地处理这一系列不同任务[70, 71]。

上述任务学习顺序的选择和安排，并没有放之四海而皆准的“金科玉律”，需要根据具体的学习任务和目标，进行针对性的调整和设计。如何在口令强度评估领域应用持续学习技术，让口令强度评估器能够同时评估不同攻击场景下的口令抗猜测性，是一个富有挑战性的课题。

3.3.4 评价指标

口令重用攻击下的口令强度评估，可以被视为一个二分类问题，即用户口令是否能在一定猜测数目下被口令重用攻击破解。一般认为，口令重用攻击主要被用于在线猜测攻击，我们此处与现有的口令重用攻击相关研究保持一致，选择 1000 次猜测作为口令重用攻击的猜测数目阈值。那么，在这个二分类问题上，对于基于口令重用的口令强度评估器，其准确性可以用平衡准确性（Balanced Accuracy）[72–74] 描述：

$$BA = (\frac{p'}{p} + \frac{n'}{n})/2 = (TPR + TNR)/2, \tag{3.6}$$

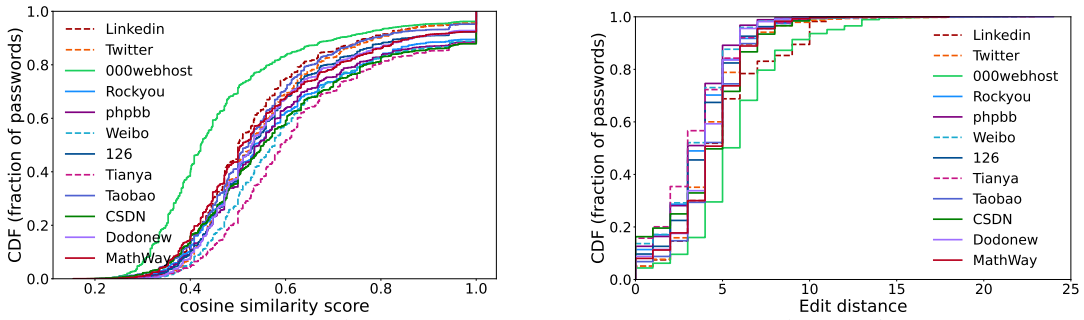
其中，TPR 表示真阳性率 (评估器所成功识别的不安全口令), TNR 则表示假阳性率 (评估器所成功识别的安全口令)。口令是否安全，则由口令重用攻击来具体确定。在评价基于口令重用的强度评估器准确性时，使用平衡准确性的好处在于其对样本均衡性不敏感。在口令重用攻击下的强度评估中，被重用攻击破解的口令（不安全口令）只占据所有口令中的一小部分，相当一部分口令是不会被破解的（往往不会超过 50%，最低可达 10% 左右）。假设仅有 10% 的口令被破解，那么一个毫无作用的、将所有口令视为安全口令的评估器将会获得 90% 的准确率，这显然是不合理的。所以在本章里，我们主要使用平衡准确性来评价口令强度评估器在口令重用攻击下的准确性。

除了使用平衡准确率以外，本文还将使用 AUC、F1-Score[73, 75] 等常用指标来衡量口令重用攻击下强度评估器的准确性，以便更完整、准确地呈现不同口令强度评估器之间的性能差异。相关实验结果参见附录第四节

第四节 基于持续学习的多功能口令强度评估器设计框架

本节中，我们主要介绍两种口令重用行为的特点以及从中得出的一些关键结论，并通过这些结论，敲定将持续学习技术应用于口令重用模型的诸多细节。通过持续学习技术，我们将把口令重用模型改造为多功能的口令强度评估器，并介绍其工作流程。

3.4.1 两种口令重用行为的特点



二元余弦相似度　　编辑距离

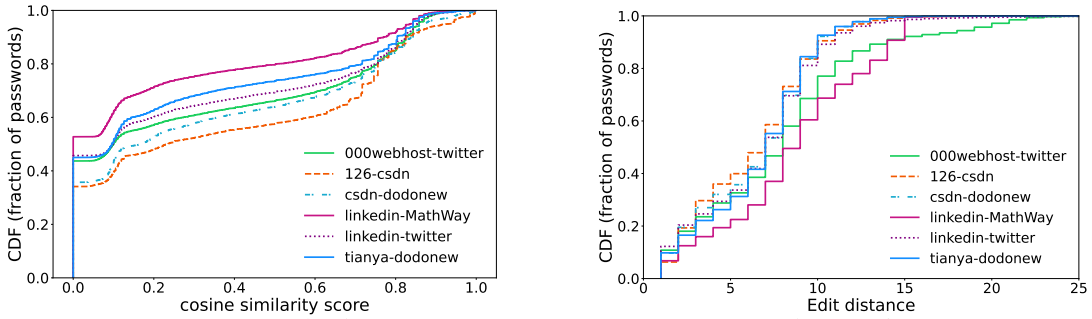
图 3.1 各数据集里抽样的用户口令与流行口令的相似度分布。

首先，我们观察真实数据集里用户口令与流行口令之间的相似度。对于每个数据集，我们将其中频数最高的前 1000 个口令作为流行口令，然后从这些流行口令之外，抽样 10^4 个用户口令，然后计算这些用户口令和流行口令的相似度。对于两个口令之间的相似度计算，我们采用二元余弦相似度和编辑距离作

为相似度的指标。其中，二元余弦相似度数值越高，两个口令越相似；编辑距离越大，则两个口令越不相似。在计算用户口令与流行口令的相似度时，我们从 1000 个流行口令中，选择与用户口令相似度最高的流行口令，并记录其相似度分数作为用户口令与流行口令的相似度。实验结果如图3.1所示，我们发现 28.7%~56.6% 的用户口令与流行口令高度相似。具体来说，我们采用现有研究所采取的相似度阈值，如果两个口令的二元余弦相似度高于 0.5，或编辑距离小于 5 时，认为两个口令是高度相似的。在这两个阈值下，与流行口令高度相似的用户口令的占比分别为 56.6% 和 28.7%。基于上述实验结果，我们可以得到如下结论：

结论 3.1 用户的口令重用行为不仅局限于重用其它账户的口令，还可以被扩展到对流行口令的重用上。

先前，我们在第二章中初步通过实验观察总结出了上述规律。现在，我们进一步通过更加丰富的口令数据集，以及更加具体的实验数据，将该规律加以确认，为持续学习的引入和多功能口令强度评估器的设计打下更牢固的基础。



二元余弦相似度　　编辑距离

图 3.2 同一用户在两个不同账户的口令之间的相似度分布。

接下来，我们观察两种重用行为，即基于流行口令和旧口令两种口令重用行为之间存在的差别。我们从配对得到的口令重用数据集里面，计算得到了不同数据集间，同一用户在两个不同账户中所用口令（姊妹口令）的相似度分布。实验结果如图3.2所示，可以发现用户姊妹口令之间的相似度，相较用户口令与流行口令之间的相似度而言，要更低一些。具体而言，用户姊妹口令之间的平均二元余弦相似度为 0.302，平均编辑距离则为 6.89；而用户口令与流行口令之间的平均二元余弦相似度为 0.56，平均编辑距离则为 4.07。上述结果说明，基于流行口令和旧口令的两种口令重用行为之间，存在复杂度上的显著差别：

结论 3.2 用户重用旧口令的行为，比用户重用流行口令的行为更加复杂。

对于这些统计结果所体现出的两种重用行为差别，我们进行了显著性检验，实验结果均为 $p-value << 0.05$，说明上述结论在统计上是显著的。现在，基于3.1和3.2两个结论，我们继续总结得到第三条结论：

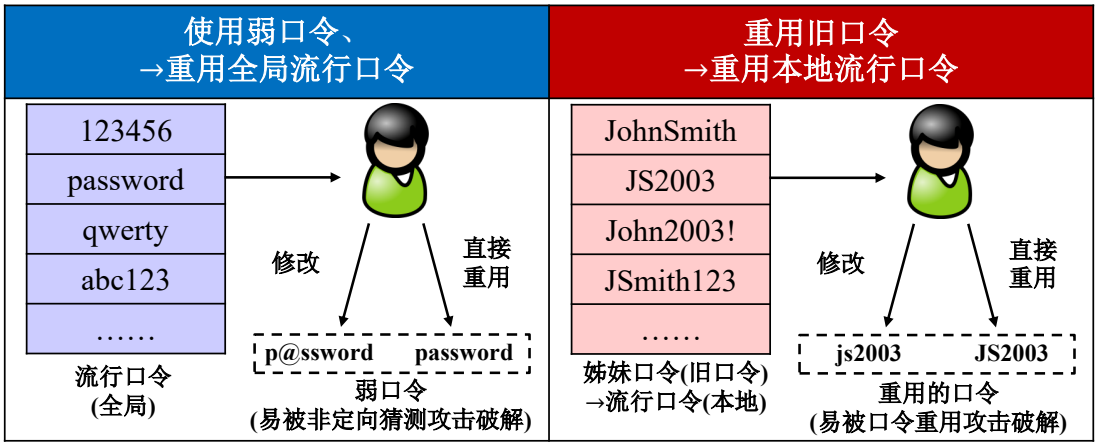


图 3.3 使用弱口令可以视为重用流行口令，将使用弱口令的行为转化为口令重用行为的一个种类，与重用旧口令的行为进行统一。

总结。结论3.1和3.2为持续学习在口令强度评估中的应用奠定了基础。用户使用弱口令的行为和重用旧口令的行为，可以视为口令重用行为的两个种类，即全局口令重用行为和本地口令重用行为。如图3.3所示，我们可以将用户使用弱口令和重用旧口令的两大脆弱口令行为，通过重用行为进行形式上的统一。其中，使用弱口令的行为即为重用流行口令的行为，流行口令是在用户群体层面上全局流行的口令，所以我们可以将其称为全局重用行为；而旧口令可以视为一种仅在用户本人层面流行的口令（例如，某名为张三的用户可能较多使用”zhangsan123” 作为自己的口令），所以重用旧口令的行为可以称为本地重用行为。在上述结论的支撑下，我们可以使用口令重用模型同时评估口令在非定向猜测攻击、口令重用攻击这两种攻击下的抗猜测性，并且设计持续学习方案，让口令重用模型同时捕捉基于流行口令和旧口令的两种口令重用行为。相关内容我们将在下一节进行介绍。本文还使用其它相似度指标对上述结论进行了进一步验证，参见附录第三节。

3.4.2 持续学习关键设计细节的理论分析

持续学习将深度学习模型在多个相关的任务上，按照预先确定的任务顺序进行训练，实现任务间的知识迁移，让模型同时对多个不同任务进行处理和建模[70, 76]。该技术已被广泛应用于自然语言处理[77, 78] 和计算机视觉[79, 80] 等多个不同领域。

持续学习技术的设计要点在于，合理排定任务的训练顺序，并采用模型参数冻结策略，以便在学习新知识、遗忘旧知识之间取得平衡[70, 71, 81]。由于本研究需要口令重用模型同时学习本地重用和全局重用两种行为，在后续的分析中，根据它们的训练先后顺序，本文将先进行训练和后进行训练的任务分别称为初始任务和后续任务。

此处，本研究假设模型在初始任务上的训练结束时，模型已经收敛到局部最优。本文将初始任务上预训练得到的模型、后续任务上微调后得到的模型分别记为 $\boldsymbol{\theta}_{\text{pt}}, \boldsymbol{\theta}_{\text{full}} \in \mathbb{R}^d$。受机器学习理论研究的启发[82–86]，我们从泛化风险和遗忘风险两个角度，在理论上分析如何对口令强度评估中应用持续学习技术。具体来说，泛化风险 $\mathcal{R}_{\text{gen}}$ 刻画了模型在初始任务上训练之后，在后续任务上进行训练时的泛化能力，遗忘风险 $\hat{\mathcal{R}}_{\text{forget}}$ 则衡量了在后续任务上微调后，模型在初始任务上的性能退化程度。下面我们对这些风险进行形式化定义，并给出对二者进行最小化的基本原则。

定义 3.3 泛化风险。对于定义在微调后模型 $\boldsymbol{\theta}_{\text{full}}$ 上的后续任务假设 $h_t : \mathcal{X} \to \mathcal{Y}$ 和有界损失函数 $\ell_t : \mathcal{Y} \times \mathcal{Y} \to [0, b]$，泛化风险定义为经验损失：

$$\hat{\mathcal{L}}_t(h_t) = \frac{1}{n} \sum_{i=1}^{n} \ell_t(h_t(x_i), y_i) \tag{3.7}$$

与其期望 $\mathcal{L}_t(h_t) = \mathbb{E}_{\mathcal{D}_t}[\ell_t(h_t(x), y)]$ 之间的偏差，即

$$\mathcal{R}_{gen}(h) = \mathbb{E}_{\mathcal{D}_t}[\ell_t(h_t(x), y)] - \frac{1}{n} \sum_{i=1}^{n} \ell_t(h_t(x_i), y_i), \tag{3.8}$$

其中 $\mathcal{D}_t$ 表示后续任务在 $\mathcal{X} \times \mathcal{Y}$ 上的未知数据分布，$\{(x_i, y_i)\}_{i=1}^{n}$ 为从 $\mathcal{D}_t$ 中抽取的 n 个样本。泛化风险可由 *Rademacher* 复杂度给出上界（见引理*3.4*）。

经验遗忘风险。对于预训练模型，其在后续任务上微调前后的初始任务经验损失变化量，即为经验遗忘风险：

$$\hat{\mathcal{R}}_{forget}(s, t) = \frac{1}{n} \sum_{i=1}^{n} \left(\ell_s(h_t(x_i), y_i) - \ell_s(h_s(x), y) \right). \tag{3.9}$$

其中 ℓ_s 为预训练模型在初始任务上的损失函数。

定义3.3中，式3.8将泛化风险定义为经验损失相对于其期望的偏离，数值越小表示泛化能力越好；式3.9则给出了一个可在实际训练数据上计算得到的遗忘风险估计量，用于刻画微调前后初始任务性能的变化。由于真实遗忘风

险 $\mathcal{R}_{\text{forget}}(s,t)=\mathbb{E}_{\mathcal{D}_s}[\ell_s(h_t(x),y)]-\mathbb{E}_{\mathcal{D}_s}[\ell_s(h_s(x),y)]$ 依赖于未知的初始任务分布 $\mathcal{D}_s$，后续分析中的遗忘风险，如无特别说明均指经验遗忘风险。

下面，我们通过引理3.4给出基于 Rademacher 复杂度的泛化风险上界。

引理 3.4 设损失函数有界 $|\ell_t(h_t(x),y)|\le b$ 且满足 ρ_1-*Lipschitz* 条件，假设类 $\mathcal{H}_t$ 的经验 *Rademacher* 复杂度为 $\hat{\mathfrak{R}}(\mathcal{H}_t)=\mathbb{E}\big[\sup_{h_t\in\mathcal{H}_t}\frac{1}{n}\sum_{i=1}^{n}\sigma_i h_t(x_i)\big]$（其中 σ_i 均匀取自 $\{-1,1\}$）。对于定义*3.3*中的泛化风险，以下不等式以概率 $\ge 1-\eta$ 对所有 $h_t\in\mathcal{H}_t$ 成立：

$$\mathcal{R}_{gen}(h_t)\le 2\rho_1\hat{\mathfrak{R}}(\mathcal{H}_t)+3b\sqrt{\frac{\ln(2/\eta)}{n}} \tag{3.10}$$

上式右端称为泛化风险的上界。若 t_A 和 t_B 分别为持续学习的初始和后续任务，则将该上界记为 $\mathcal{G}_{t_A\to t_B}$。

证明. 令 $\tilde{h}_t(x)=h_t(x)/b$，由 Rademacher 复杂度上界可得

$$\mathcal{R}_{gen}(h_t)=\mathbb{E}_{\mathcal{D}_t}\big[\ell_t(\tilde{h}_t(x),y)\big]-\frac{1}{n}\sum_{i=1}^{n}\ell_t(\tilde{h}_t(x_i),y_i)\le 2\hat{\mathfrak{R}}(\ell(\mathcal{H}_t/b))+3\sqrt{\frac{\ln(2/\eta)}{n}} \tag{3.11}$$

两边同乘 b 并利用 Lipschitz 条件，即得式3.10。 □

接下来，本文通过定理3.5建立了给定初始任务和后续任务时泛化风险和遗忘风险的上界，为初始与后续任务的选择提供指导。

定理 3.5 设 $t_A\to t_B$ 表示以 t_A 和 t_B 分别为初始和后续任务的持续学习情形。对于泛化风险，若两任务假设类满足 $\mathcal{H}_{t_A}\supseteq\mathcal{H}_{t_B}$ 且样本量满足 $n_{t_A}\le n_{t_B}$，则其上界满足 $\mathcal{G}_{t_A\to t_B}\le\mathcal{G}_{t_B\to t_A}$。对于遗忘风险，若初始任务损失函数关于模型参数 $\boldsymbol{\theta}$ 的梯度和 *Hessian* 矩阵在预训练模型 $\boldsymbol{\theta}_{\text{pt}}$ 附近满足 $\|\nabla\mathcal{L}_s(\boldsymbol{\theta}_{\text{pt}})\|=0$ 以及 $\|\nabla^2\mathcal{L}_s(\boldsymbol{\theta})-\nabla^2\mathcal{L}_s(\boldsymbol{\theta}_{\text{pt}})\|\le\rho_2\|\boldsymbol{\theta}-\boldsymbol{\theta}_{\text{pt}}\|$（其中 ρ_2 为正常数），则

$$\hat{\mathcal{R}}_{forget}(t_A,t_B)\le\frac{1}{2}\Big(\lambda_{\max}\nabla^2\mathcal{L}_s(\boldsymbol{\theta}_{\text{pt}})\Big)\|\boldsymbol{\delta}\|^2 \tag{3.12}$$

其中 $\lambda_{\max}$ 为 *Hessian* 矩阵的最大特征值，$\boldsymbol{\delta}=\boldsymbol{\theta}-\boldsymbol{\theta}_{\text{pt}}$ 为模型参数在 $\boldsymbol{\theta}_{\text{pt}}$ 附近的更新量。

证明. 对于泛化风险，由 $\mathcal{H}_B\subseteq\mathcal{H}_A$ 可知经验 Rademacher 复杂度满足 $\hat{\mathfrak{R}}(\mathcal{H}_B)\le\hat{\mathfrak{R}}(\mathcal{H}_A)$，由引理3.4得

$$\mathcal{G}_{t_A\to t_B}-\mathcal{G}_{t_B\to t_A}=2\rho_1(\hat{\mathfrak{R}}(\mathcal{H}_B)-\hat{\mathfrak{R}}(\mathcal{H}_A))+3b(\ln(2/\eta))^{\frac{1}{2}}(n_B^{-\frac{1}{2}}-n_A^{-\frac{1}{2}})\le 0, \quad (3.13)$$

即存在 $\mathcal{G}_{t_A\to t_B}\le\mathcal{G}_{t_B\to t_A}$ 的关系。

对于遗忘风险，利用 Taylor 展开和中值定理，对初始任务损失函数存在某 $a\in(0,1)$ 使得

$$\begin{aligned}\hat{\mathcal{R}}_{forget}(t_A,t_B)&=\mathcal{L}_s(\theta_{\text{pt}}+\boldsymbol{\delta})-\mathcal{L}_s(\theta_{\text{pt}})=\nabla\mathcal{L}_s(\theta_{\text{pt}})^\top\boldsymbol{\delta}+\frac{1}{2}\boldsymbol{\delta}^\top\nabla^2\mathcal{L}_s(\boldsymbol{\theta}_{\text{pt}}+a\boldsymbol{\delta})\boldsymbol{\delta}\\&=\frac{1}{2}\boldsymbol{\delta}^\top\nabla^2\mathcal{L}_s(\boldsymbol{\theta}_{\text{pt}})\boldsymbol{\delta}+\frac{1}{2}\boldsymbol{\delta}^\top\left(\nabla^2\mathcal{L}_s(\boldsymbol{\theta}_{\text{pt}}+a\boldsymbol{\delta})-\nabla^2\mathcal{L}_s(\boldsymbol{\theta}_{\text{pt}})\right)\boldsymbol{\delta}\\&\le\frac{1}{2}\left((\boldsymbol{\delta}U^\top)\Lambda(U^\top\boldsymbol{\delta})+\rho_2\|\boldsymbol{\delta}\|^3\right)\le\frac{1}{2}\left(\lambda_{\max}\nabla^2\mathcal{L}_s(\boldsymbol{\theta}_{\text{pt}})+\rho_2\|\boldsymbol{\delta}\|\right)\|\boldsymbol{\delta}\|^2\\&\approx\frac{1}{2}\left(\lambda_{\max}\nabla^2\mathcal{L}_s(\boldsymbol{\theta}_{\text{pt}})\right)\|\boldsymbol{\delta}\|^2\end{aligned} \quad (3.14)$$

其中第三行来自 Hessian 矩阵的正交分解 $\nabla^2\mathcal{L}_s(\boldsymbol{\theta}_{\text{pt}})=U\Lambda U^T$，$\Lambda=\text{diag}(\lambda_1,\ldots,\lambda_d)$。□

定理3.5为持续学习中初始、后续任务的选择提供了参考，即选择复杂任务作为初始任务、简单任务作为后续任务，将可以在一定程度上同时降低泛化风险和遗忘风险的上界。首先，对于泛化风险，当初始任务 t_A 更复杂时，上界满足 $\mathcal{G}_{t_A\to t_B}<\mathcal{G}_{t_B\to t_A}$，表明后续任务上的过拟合风险更低。其次，经验遗忘风险 $\hat{\mathcal{R}}_{forget}$ 与预训练模型和微调模型之间参数位移量 $\|\boldsymbol{\delta}\|$ 的平方成正比。而在更复杂的初始任务上进行预训练，则可以一定程度上限制了微调过程中 $\|\boldsymbol{\delta}\|$ 的大小，进而降低模型在初始任务上的遗忘风险[71, 87–90]。

上述分析为初始、后续任务的选择提供了理论遵循。下面，本文继续分析模型在后续任务的训练过程中冻结部分参数的有效性。

定理 3.6 设初始任务 t_A 上训练模型得到的模型 $\boldsymbol{\theta}_{\text{pt}}$ 被划分为冻结部分 *(F)* 和未冻结部分 *(UF)*，其在后续任务 t_B 上微调时，对应的参数更新 $\boldsymbol{\delta}$ 分别为 $\boldsymbol{\delta}^{\text{full}}=(\boldsymbol{\delta}_F^{\text{full}},\boldsymbol{\delta}_{\text{UF}}^{\text{full}})$（不使用参数冻结，进行全参数微调）和 $\boldsymbol{\delta}^{\text{frz}}=(\mathbf{0},\boldsymbol{\delta}_{\text{UF}}^{\text{frz}})$（冻结部分参数之后进行微调）。采用全量微调和部分冻结微调时，泛化风险的上界满足 $\mathcal{G}^{\text{frz}}<\mathcal{G}^{\text{full}}$（假设 $\mathcal{H}_{\text{frz}}\subseteq\mathcal{H}_{\text{full}}$），而初始任务遗忘风险的差值为

$$\hat{\mathcal{R}}^{\text{full}}_{forget}(t_A,t_B)-\hat{\mathcal{R}}^{\text{frz}}_{forget}(t_A,t_B)$$
$$=\frac{1}{2}(\boldsymbol{\delta}^{\text{full}}_F)^\top H_{F,F}\boldsymbol{\delta}^{\text{full}}_F+(\boldsymbol{\delta}^{\text{full}}_F)^\top H_{F,\text{UF}}\boldsymbol{\delta}^{\text{full}}_{\text{UF}}+\frac{1}{2}\left[(\boldsymbol{\delta}^{\text{full}}_{\text{UF}})^\top H_{\text{UF,UF}}\boldsymbol{\delta}^{\text{full}}_{\text{UF}}-(\boldsymbol{\delta}^{\text{frz}}_{\text{UF}})^\top H_{\text{UF,UF}}\boldsymbol{\delta}^{\text{frz}}_{\text{UF}}\right], \tag{3.15}$$

其中 *Hessian* 矩阵 $H_S=\nabla^2\mathcal{L}_S(\boldsymbol{\theta}_{\text{pt}})=\begin{bmatrix}H_{F,F} & H_{F,\text{UF}}\\ H_{\text{UF},F} & H_{\text{UF,UF}}\end{bmatrix}$。

证明. 对于泛化风险，其证明与定理3.5类似，此处略去。对于遗忘风险，将定理3.5分别应用于全量微调和部分冻结微调两种情形，可得

$$\hat{\mathcal{R}}^{\text{full}}_{forget}(t_A,t_B)=\frac{1}{2}(\delta^{\text{full}})^T H_S\delta^{\text{full}},\qquad \hat{\mathcal{R}}^{\text{frz}}_{forget}(t_A,t_B)=\frac{1}{2}(\delta^{\text{frz}})^\top H_S\delta^{\text{frz}}. \tag{3.16}$$

展开全量微调的二次型，得到

$$(\boldsymbol{\delta}^{\text{full}})^\top H_S\boldsymbol{\delta}^{\text{full}}=(\boldsymbol{\delta}^{\text{full}}_F)^\top H_{F,F}\boldsymbol{\delta}^{\text{full}}_F+2(\boldsymbol{\delta}^{\text{full}}_F)^\top H_{F,\text{UF}}\boldsymbol{\delta}^{\text{full}}_{\text{UF}}+(\boldsymbol{\delta}^{\text{full}}_{\text{UF}})^\top H_{\text{UF,UF}}\delta^{\text{full}}_{\text{UF}}. \tag{3.17}$$

冻结情形仅保留最后一项。将两式相减并除以 2，即得式3.15中的差值。 □

定理3.6揭示了采用带参数冻结的部分微调的必要性和有效性。具体而言，为最小化遗忘风险，被冻结的权重应当与初始任务强相关（即 $H_{F,F}$ 对角线元素尽可能大），且与后续任务相关性尽可能低（即 $H_{F,\text{UF}}$ 的元素尽可能小）。关于选择冻结参数的具体细节，本文将在下一节中阐述。

3.4.3 用持续学习改造口令重用模型

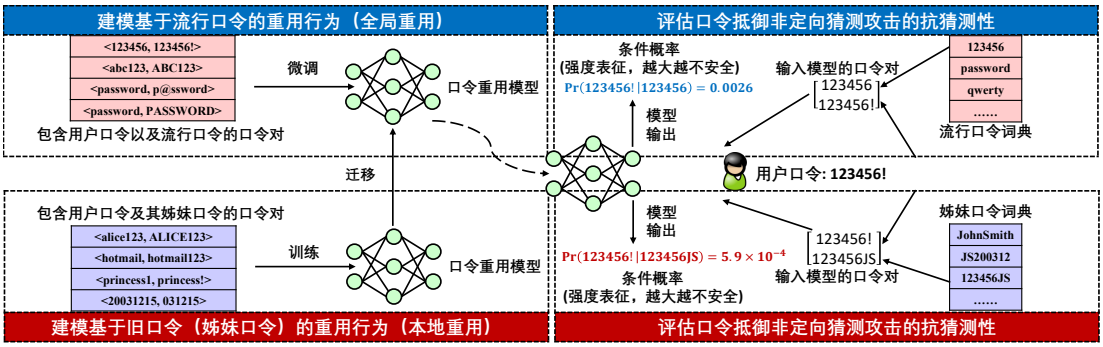


图 3.4 利用持续学习技术，将口令重用模型改造为多功能口令强度评估器。

现在，基于本文在第3.4.1节中对用户两类口令重用行为得出的两条结论，以

及在第3.4.2节中进行的理论分析，本文提出 VersaPSE (Versatile Password Strength Evaluation) 多功能口令强度评估框架。该框架能够使用持续学习技术，让口令重用模型同时准确建模基于流行口令、基于旧口令的口令重用行为，进而将现有的口令重用模型改造为多功能的口令强度评估器。

首先，设计持续学习方案的要解决的第一个设计要点是，选择合适的初始任务和后续任务，规定持续学习中不同任务的训练顺序。在定理3.5中，本文通过理论分析，指出应当选择更复杂的任务作为初始任务、相对简单的任务作为后续任务，进而降低模型在持续学习中的遗忘风险和泛化风险。在此基础上，回顾本文先前提出的结论3.2，该结论指出：基于旧口令的重用行为（本地重用行为）比基于流行口令的重用行为（全局重用行为）要更加复杂。综合上述理论分析和对用户重用行为的统计和观察，为了让模型能够获取更加丰富的先验知识，VersaPSE 框架选取更加复杂的本地重用行为作为初始任务，在重用模型能够捕捉本地重用行为后，再将模型迁移到复杂度较低的全局重用行为上进行训练。这样设计的主要考虑是，本地重用行为本身的复杂性，能够让模型学习到更丰富的重用知识，从而将这些知识进行迁移，以便更好建模相对简单的全局重用行为上；而全局重用行为本身的特点是，其包含口令数据集里的流行口令和用户口令，能够为本地重用行为提供更加广阔的口令语义与分布知识，进而在持续学习后让模型更好建模本地重用行为。

然后，在上述技术路线的基础上，还需要解决持续学习方案的另一个要点，即选择持续学习过程中所需要冻结/微调的模型权重。在持续学习中冻结部分模型权重的意义在于，模型在后续任务上进行微调时，其参数会进一步得到更新，进行部分权重冻结有助于保护模型在初始任务上学习到的先验知识，避免这些知识在微调过程中被稀释、遗忘。因此，根据定理3.6中的理论分析，我们在 VersaPSE 中所提出的冻结策略原则是：在微调时，冻结与口令重用行为模式相关的模型权重，微调与口令语义捕捉相关的模型权重。这样设计的好处是，重用模型能够在本地重用行为中学习复杂的重用行为模式，那么冻结相关权重后，模型得以在后续微调中保留相关的先验知识；模型在全局重用行为中进一步学习更广阔的口令语义信息与分布特征，那么微调相关权重，有助于模型进一步捕捉口令语义信息。

当然，需要注意的是，不同的口令重用模型结构多样，权重冻结的选择需要具体情况具体分析，这里我们以 Pass2Path 模型[47] 作为样例，介绍如何按照模型中不同组成部分的功能来选择需要冻结和微调的模型权重。如图3.5所示，

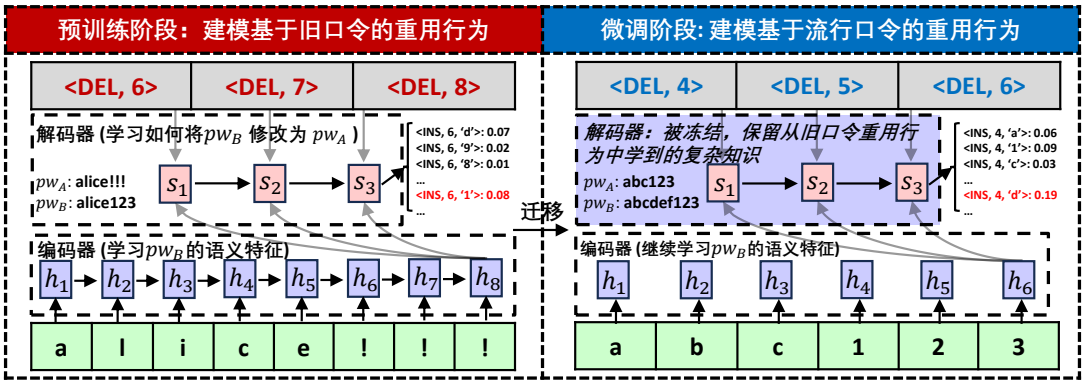


图 3.5 VersaPSE 的持续学习过程里，模型权重冻结和微调的选择原则（以 Pass2Path 为例）。

Pass2Path 模型的结构中，编码器用于学习给定口令（旧口令或流行口令）的语义特征，解码器则根据编码器得到的语义特征，输出对该口令进行编辑的一系列编辑操作。那么，由于解码器直接学习用户对口令的复杂修改行为，其模型参数将在学习流行口令重用行为时进行冻结；而编码器学习给定口令的语义特征，其参数将在全局重用行为上进行微调，以便学习流行口令中更广阔的语义特征。除了 Pass2Path 模型以外，KNNGuess、PointerGuess 等模型的参数冻结相关内容将在第3.5.1节中进行介绍。

在敲定持续学习方案中初始任务与后续任务、模型权重的冻结这两大关键细节之后，现在我们完整地介绍 VersaPSE 框架下，的训练和工作流程。在训练过程中，首先让口令重用模型在本地重用行为上进行训练，使其捕捉复杂的口令重用行为特征，然后冻结其中与重用行为相关的模型权重，将模型迁移到全局重用行为上进行训练，让模型中与口令语义信息、分布特征相关的部分得到进一步微调。在这一持续学习过程结束后，口令重用模型得以同时准确建模本地重用行为和全局重用行为，进而可以评估口令在非定向猜测攻击和口令重用攻击下的抗猜测性。

接下来，在评估口令抗猜测性时，如图3.4所示，首先我们准备流行口令词典和用户姊妹口令词典，在两个词典中分别寻找与待评估的用户口令 pw_{test} 相似度最高的流行口令 pw_{pop} 和姊妹口令 pw_{sis}，形成两个口令对 $< pw_{test}, pw_{pop} >$ 和 $< pw_{test}, pw_{sis} >$。然后，根据这两个口令对，口令重用模型分别计算两个条件概率 $\Pr(pw_{test}|pw_{pop})$ 和 $\Pr(pw_{test}|pw_{sis})$，前者作为口令在非定向猜测攻击下的强度表征，后者作为口令在重用攻击下的强度表征，条件概率越高则口令抗猜测性越差。

第五节 实验设计与结果

3.5.1 各口令重用模型的具体改造方案

本章用 VersaPSE 多功能口令强度评估方案，成功改造了现有的五款口令重用模型，对这些模型所分别采用的模型权重冻结方法和相关考量如下：

- **KNNGuess 口令重用模型**。该模型所使用的基础模型是 Transformer 模型，其主要结构包括编码器和解码器。由于编码器负责处理给定口令，解码器负责根据编码器所输出的给定口令高维表征来输出新口令，我们在持续学习过程的微调阶段冻结解码器，微调编码器。这样一来，模型的解码器可以保留从本地重用行为中学到的复杂口令重用知识，编码器则得以在全局重用行为上继续学习流行口令的语义和分布特征。
- **PointerGuess 口令重用模型**。该模型主要结构包括编码器、解码器、注意力机制和指针机制。其中，编码器学习给定口令（旧口令或流行口令）中的语义信息，注意力机制为编码器提供的语义信息分配注意力权重；指针机制则根据给定口令的语义信息，决定在新口令中是否保留、何时保留给定口令中的部分字符，解码器负责逐字符地输出最终得到的新口令。那么，根据我们提出的 VersaPSE 持续学习原则，在对 PointerGuess 进行改造时，我们在持续学习时冻结其指针机制和解码器，微调其与口令语义信息相关的注意力机制和编码器。
- **Pass2Edit 口令重用模型**。该模型的结构较为简单，其 GRU 模型既学习口令的语义特征，又学习编辑操作对口令的修改效果，并且通过两层全连接层得到当前时间步所预测的单个编辑操作。考虑到 GRU 模型的较低层学习到的是更宏观的语义特征，我们在持续学习时，冻结模型的全连接层和 GRU 的第一层，让 GRU 的后两层在全局重用行为上学习更宏观的语义信息。
- **PassBERT 口令重用模型**。该模型采用 BERT 作为基础模型，并且将口令重用行为描述为标签形式的一系列编辑操作，应用在口令的不同字符上。因此，我们选择在持续学习时冻结模型中的池化层、全连接层，以及编码器中的第一层，在全局重用行为上微调进一步编码器的剩余层。
- **Pass2Path 口令重用模型**。该模型采用 Seq2Seq 作为基本结构，其中编码器用于学习给定口令的语义特征，解码器用于根据编码器得到的语义特征，选择后续要采取的一系列编辑操作。与 PointerGuess 模型类似，我们在持

续学习时直接冻结模型的解码器，只微调编码器。

3.5.2 口令重用攻击下的口令强度评估

表 3.3 各口令强度评估器在重用攻击场景下的平衡准确性

Scenario	VersaPSE (KNNG.)*	VersaPSE (Pass2Edit)*	VersaPSE (PointerG.)*	VersaPSE (PassBERT)*	VersaPSE (Pass2Path)*	KNN-PSM	PR-PSM	BERT-PSM	Vec-PPSM
#1	0.930 (+11.8%~ +72.2%)	0.918 (+10.3%~ +70.0%)	0.867 (+4.2%~ +60.5%)	0.812 (-2.5%~ +50.3%)	0.906 (+8.9%~ +67.8%)	0.695 [0.832]	0.719	0.540 [0.679]	0.719 [0.767]
#2	0.883 (+4.0%~ +45.7%)	0.892 (+5.1%~ +47.2%)	0.864 (+1.8%~ +42.6%)	0.812 (-4.3%~ +34.1%)	0.875 (+3.1%~ +44.4%)	0.606 [0.827]	0.849	0.636 [0.840]	0.623 [0.732]
#3	0.954 (+1.1%~ +76.3%)	0.951 (+0.8%~ +75.8%)	0.938 (-0.5%~ +73.4%)	0.883 (-6.3%~ +63.3%)	0.909 (-3.6%~ +68.1%)	0.541 [0.688]	0.943	0.786 [0.856]	0.667 [0.732]
#4	0.890 (+9.3%~ +57.5%)	0.881 (+8.3%~ +56.0%)	0.861 (+5.8%~ +52.4%)	0.803 (-1.4%~ +42.1%)	0.864 (+6.1%~ +52.9%)	0.601 [0.814]	0.799	0.565 [0.788]	0.620 [0.709]
#5	0.950 (+4.5%~ +65.5%)	0.941 (+3.5%~ +63.9%)	0.927 (+2.0%~ +61.5%)	0.886 (-2.5%~ +54.4%)	0.916 (+0.7%~ +59.5%)	0.574 [0.710]	0.909	0.814 [0.886]	0.680 [0.760]
#6	0.925 (+4.0%~ +40.6%)	0.895 (+0.6%~ +36.1%)	0.879 (-1.3%~ +33.6%)	0.852 (-4.3%~ +29.5%)	0.896 (+0.7%~ +36.2%)	0.803 [0.890]	0.691	0.658 [0.754]	0.705 [0.759]
#7	0.926 (+4.2%~ +46.6%)	0.901 (+1.4%~ +42.6%)	0.865 (-2.7%~ +36.8%)	0.869 (-2.2%~ +37.6%)	0.900 (+1.2%~ +42.3%)	0.830 [0.889]	0.673	0.632 [0.699]	0.722 [0.753]
#8	0.891 (+4.5%~ +35.9%)	0.869 (+1.8%~ +32.4%)	0.860 (+0.9%~ +31.2%)	0.833 (-2.3%~ +27.0%)	0.858 (+0.6%~ +30.8%)	0.728 [0.853]	0.819	0.749 [0.844]	0.656 [0.741]
Average	0.919 (+13.0%~ +36.7%)	0.906 (+11.4%~ +34.8%)	0.883 (+8.6%~ +31.3%)	0.844 (+3.8%~ +25.6%)	0.891 (+9.5%~ +32.5%)	0.672 [0.813]	0.800	0.673 [0.793]	0.674 [0.744]

* 这里为各口令重用模型在 VersaPSE 改造后所产生的口令强度评估器。
圆括号中的百分比为本文改造得到的多功能口令强度评估器，相对现有的口令强度评估器在准确性上的提升幅度。
方括号中的实验结果为现有口令强度评估器在 Zxcvbn-PSM 的协助下，所取得的准确性。

我们将 VersaPSE 改造而成的重用口令强度评估器，与现有的口令强度评估器一道，在3.3.2中所描述的口令重用攻防场景里进行了系统性实验。考虑到我们按照现有研究的推荐设定，在口令重用攻击中混入了流行口令，而除 VersaPSE 以外的其它重用口令强度评估器无法有效捕捉流行口令的强度，我们利用 Zxcvbn-PSM 作为现有一系列重用口令强度评估器的辅助评估器。该设定与已有研究保持了一致[47]。如表3.3所示，KNN-PSM、BERT-PSM、Vec-PPSM 括号中的分数即为在 Zxcvbn-PSM 辅助下的口令强度评估平衡准确性。需要注意的是，由于 PR-PSM 本身就结合了 Zxcvbn-PSM，所以其分数直接呈现在表中，不以括号的形式表示。学术界现有口令强度评估器的具体设定参见附录第二节。

从实验结果中可以发现，通过 VersaPSE 改造得到的五个重用口令强度评估器，即 VersaPSE(KNNGuess/Pass2Edit/PointerGuess/PassBERT/Pass2Path)，均在准确性层面全面超过了前人提出的重用口令强度评估器。首先，基于相似度的

Vec-PPSM 在准确性上显著低于 VersaPSE 改造后的口令重用模型。这在很大程度上是因为 Vec-PPSM 只考虑用户口令在文本层面的词嵌入相似度[91]，而并不考虑用户重用行为的复杂性，进而难以准确将”语义上存在相似性的口令”和”重用攻击下容易被破解的口令”区分开来，无法给用户提供精确的口令强度评估。相反，我们的 VersaPSE 框架所改造得到的口令强度评估器，其本身基于口令重用模型，在经过持续学习后能够较好地对口令重用行为进行建模，进而对用户口令进行基于重用行为的强度评估。

KNN-PSM、BERT-PSM、PR-PSM 分别利用 KNNGuess、PassBERT 和 PointerGuess 生成口令猜测，进而借助这三种口令重用模型进行口令强度评估。可以看到，VersaPSE 对 KNNGuess、PassBERT、PointerGuess 的改造方法，比 KNN-PSM、BERT-PSM、PR-PSM 要有效得多。分析表3.3不难发现，VersaPSE 改造后的 KNNGuess、PassBERT、PointerGuess 在准确性上全面超越了 KNN-PSM、BERT-PSM 和 PR-PSM。这主要是因为 KNN-PSM、BERT-PSM 和 PR-PSM 利用口令猜测进行强度评估的方式存在严重的局限性，即单款口令重用模型所能破解的口令，无法覆盖所有在重用攻击下易被攻破的口令。此外，利用猜测生成的方式评估口令强度，不仅需要较高计算成本，其所提供的强度反馈同样是二值化的，只有被破解/未被破解两种反馈。VersaPSE 则巧妙地通过口令重用模型输出的条件概率作为强度表征，提供了更细粒度的口令强度反馈，有效规避了基于猜测生成评估口令强度所面临的覆盖率低、强度反馈二值化问题。

此外，VersaPSE 改造口令重用模型所得到的口令强度评估器，同样可以评估非定向猜测下的口令强度。在实验中，即便给现有的重用口令强度评估器，搭配一个准确度较高的 Zxcvbn-PSM 作为非定向猜测攻击下的强度评估辅助，VersaPSE 所改造得到的强度评估器仍然在准确性上大幅超越了现有各类方法。这一结果说明，我们在 VersaPSE 中所提出的持续学习方法论是卓有成效的，其训练得到的口令重用模型确实可以有效捕捉基于流行口令和基于旧口令的两种重用行为。有关 VersaPSE 改造口令重用模型得到的评估器在非定向猜测场景下的准确性，我们将在下一节里进行介绍。

3.5.3 非定向口令猜测攻击下的口令强度评估

参照第二章中对非定向口令猜测场景中口令强度评估的实验设定，我们使用 *WSpearman* 系数对 VersaPSE 改造得到的多功能口令强度评估器，以及现有的一系列非定向口令强度评估器，进行了准确性评估和比较。

表 3.4 非定向口令强度评估场景

#	语言 (训练集 A*)	训练集 B*	训练集 C*	测试集
1	中文 (Tianya)	Weibo	Tianya-Dodonew	126
2		Dodonew	Tianya-Dodonew	CSDN
3		Taobao	CSDN-126	Dodonew
4	英文 (Rockyou)	Phpbb	Twitter-LinkedIn	000webhost
5		Twitter	000webhost-Twitter	LinkedIn
6		000webhost	LinkedIn-MathWay	Twitter

* 训练集 A 用于训练所有只需要一个训练集的口令强度评估器，同时也用于 VersaPSE 的持续学习环节。训练集 B 用于为 fuzzyPSM 进行训练，根据其设计原理，需要使用 A、B 两个数据集才能对其进行有效训练。训练集 C 包含用户的姊妹口令对，用于训练 VersaPSE 框架下的口令重用模型。

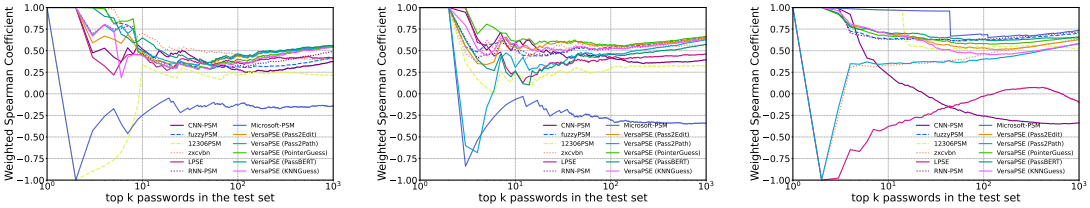
如表3.4所示，我们设计了六个非定向口令猜测下的口令强度评估场景。其中，为了说明 VersaPSE 在非定向攻击、重用攻击两个场景下均能准确评估口令强度，我们用于训练 VersaPSE 下口令重用模型的口令对和持续学习训练集，均与前文中的重用口令强度评估场景保持一致。

在上述六个非定向口令强度评估场景中，VersaPSE 所改造得到的五款多功能口令强度评估器和现有口令强度评估器的 *WSpearman* 分数如图3.6和3.7所示。此处的热力图描述的是 VersaPSE（PointerGuess）和各现有口令强度评估器的 *WSpearman* 相对值。可以看到，VersaPSE 改造后得到的五款多功能口令强度评估器均实现了现有口令强度评估器中的先进水平，其中 VersaPSE(PointerGuess) 在所有标志性口令排名位置的平均表现最佳。而其余的四款多功能口令强度评估器，即 VersaPSE(KNNGuess/Pass2Edit/PassBERT/Pass2Path)，同样在各标志性口令排名位置的平均分数上，实现了对现有一系列非定向口令强度评估器的超越。其它各多功能口令强度评估器的热力图参见附录第五节。

第六节 本章小结

在本章中，我们通过对大规模真实口令数据集进行观察，分析并总结了基于流行口令和基于旧口令两种重用行为的特点和区别，得出了三条结论。通过这三条结论，我们进一步设计了对口令重用模型进行持续学习的具体技术路线，从而提出 VersaPSE 多功能口令强度评估器设计框架。该框架能够通过持续学习技术，让口令重用模型同时捕捉两种口令重用行为，从而系统性地将其改造成多功能口令强度评估器。

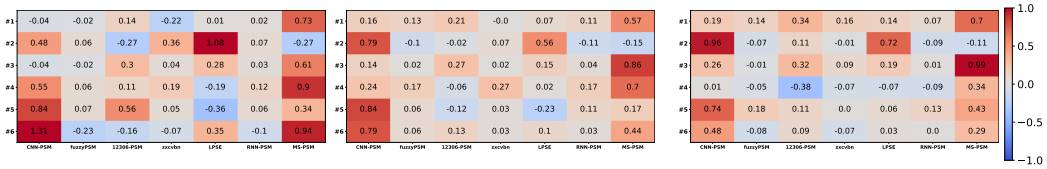
通过 VersaPSE 框架，我们将 KNNGuess、PointerGuess、Pass2Edit、PassBERT 和 Pass2Path 口令重用模型分别改造成了五款多功能口令强度评估器。实验结果表明，这五款多功能口令强度评估器在口令重用攻击下的口令强度评估场景中，



(a) #1, Tianya→126 (b) #2, Tianya→Dodonew (c) #3, Tianya→CSDN

(d) #4, Rockyou→Linkedin (e) #5, Rockyou→Twitter (f) #6, Rockyou→000webhost

图 3.6 抵御先验知识充分攻击者的实验结果。



(a) Rank= 10. (b) Rank= 10^2. (c) Rank= 10^3.

图 3.7 VersaPSE(PointerGuess) 在不同标志性排名位置的相对 *WSpearman* 分数热力图。

准确性大幅超越了现有的重用口令强度评估器；在非定向口令强度评估场景中，同样达到了现有一系列非定向口令强度评估器中的先进水平。我们相信，本章的工作将为多功能口令强度评估器的设计奠定新的基础，并且为口令重用模型在口令强度评估中的应用做出贡献。

第四章　深度学习技术在蜜口令库方案中的应用

我们将在本章里介绍如何利用深度学习技术设计蜜口令库方案。本章利用量化变分自编码器设计了蜜口令库方案，其能够通过有限公开真实口令库中的部分口令分布、语义特征信息，相对现有的蜜口令库方案实现更高的安全性。同时，公开这一部分信息，不会让攻击者在区分攻击中取得额外的区分优势。

第一节　研究背景

随着互联网的不断发展，用户所需要维护的账户口令数量不断上升（互联网用户平均有 80-107 个账户[92, 93]），而用户用于记忆口令、管理口令的时间精力却极其有限。这导致用户为了更好地记忆口令，在设置口令时存在种种脆弱行为，比如选择易于记忆的流行口令[68]、基于个人信息的口令[22] 和跨账户重用的口令[22]。攻击者可以通过种种方式，对这类脆弱的口令行为进行建模，进行针对性的口令猜测攻击 [22, 41, 58]，导致巨大的财产损失和安全隐患。

为了提高口令的可用性和安全性，口令管理器（Password Manager, PM）提供了一种简单的口令管理方案，能够大大减轻用户记忆口令的负担[94]。在口令管理器中，用户只需要记忆特定的密钥（如用户设置的主口令），即可存储和访问整个口令库。此外，口令管理器还可以提供强口令生成、自动填充、跨设备口令同步等辅助功能，给用户带来了诸多便利。在实际场景中，口令管理器因其实用性得到了广泛的应用：浏览器自带的“密码管理器”，以及手机系统中常见的“密码保险箱”，都拥有大量用户、备受人们青睐。

然而，口令管理器在为用户提供便利的同时，也带来了诸多安全隐患。口令管理器只需正确输入主口令即可访问所有账户的口令，其安全性高度依赖于主口令的抗猜测攻击的强度。事实上，用户在选用主口令时，仍然存在诸如口令重用、弱口令、流行口令等脆弱口令行为。Security.org 的调研人员[95] 指出，2023 年中，高达 28% 的用户将在其它账户使用过的口令直接作为口令管理器的主口令，一旦使用相同口令的其它账户发生泄露，口令管理器的主口令极易被攻破。尽管口令管理器的服务提供方可以设置种种安全机制（最大口令输入错误次数，账户冻结，登录限制等），但研究表明[22]，即便存在登录次数上限等安全机制，攻击者成功破解用户口令的概率仍可高达 60%，这无疑给口令管理器的安全性带来了极大威胁。

除此之外，由于口令管理器提供跨设备口令同步、云存储口令等功能，用户

的口令将会以密文形式存储在服务提供商的远程服务器中，而这些远程服务器存在被入侵和攻破的风险。主流的口令管理器，如 LastPass、1Password、KeePass 和一些浏览器内置的口令管理器等都存在严重的安全漏洞，并且遭到过多次攻击[11, 96]。

2021 年 4 月，攻击者利用 Passwordstate 客户端的漏洞，窃取了域名和用户口令库密文等数据[97]。2022 年，LastPass 连续两次被攻击者入侵，超过 2500 万用户的口令库密文、用户名和电话号码等信息遭到泄露[98]。在口令库密文泄露后，攻击者可以在本地对主口令进行猜测，猜测正确后即可对口令库密文进行解密，恢复出用户的口令。此时，攻击者的猜测能力不再受到服务器安全机制的限制，而只会受到主口令复杂度、口令猜测软件效率（如 HashCat[52]）和算力（如 RTX 4090 GPU[99]）的影响。随着口令猜测软件的能力不断提升[13, 39, 100] 和计算设备性能的提高[99, 101]，即便口令管理器以密文形式存储口令库，一旦其遭到泄露，安全性仍会受到极大威胁。因此，在口令库密文泄露后，抵抗攻击者的离线口令猜测攻击，保护用户口令的安全性，提高口令管理器的抗泄露能力，具有十分重要的研究价值和现实意义。

4.1.1 研究动机

迄今为止，学术界已经提出了一系列蜜口令库方案，并且均声称其达到了较高的安全性，能够抵御攻击者实施的区分攻击。但是，这些蜜口令库方案迄今未被工业界的口令管理器等场景中得到实际部署和应用。这在一定程度上是因为，学术界声称安全的这些蜜口令库方案，往往在提出后数年之内就被新的、更先进的区分攻击所攻破。例如，Chatterjee 等人[34] 在 2015 年所提出的 NoCrack 蜜口令库方案，声称其能抵御利用机器学习等技术的区分攻击，但 Golla 等人在 2016 年就提出基于口令分布的区分攻击方案，将其安全性削弱了将近 50%[35]；Golla 等人同样在 2016 年提出了声称能抵御各类区分攻击的蜜口令库方案，但该方案在 2019 年和 2021 年被 Cheng 等人根据编码器、自适应机制设计的区分攻击攻破[36, 37]；Cheng 等人在 2021 年提出的蜜口令库方案，声称其能通过使用“最佳”参数的马尔科夫口令模型抵御各类区分攻击，而 Hou 等人[49] 在 2025 年通过优化现有的区分攻击方案，即可将 Cheng 等人设计的蜜口令库方案安全性削弱 30% 以上。

到底是什么原因，导致学术界在蜜口令库方案领域的研究陷入“提出防御机制-被快速攻破”的循环中？究其根本，现有的蜜口令库方案在进行安全性评

估时，所使用的攻防场景和现实场景存在较大的差别。在 2025 年以前，所有蜜口令库方案的安全性测试完全依赖于迄今为止唯一公开可获取的、口令库数量仅为 276 个的 Pastebin 数据集[34–37, 49]。由于该口令库数据集较小，口令总数较低，在该数据集上进行安全分析所得到的结论在可信度上也打了折扣[102]。比如，Golla 等人[35] 和 Cheng 等人[36] 的蜜口令库方案，其所采用的一系列超参数均依赖于在 Pastebin 上进行安全性分析所得到的实验结果，在使用不同训练集、不同测试集时，其安全性会发生较大偏差[49, 102]。没有新的、更大规模的数据集，蜜口令库方案在现实场景中的安全性分析就无从谈起。

在 2025 年，Cheng 等人[102] 首次使用一系列大规模口令数据集，通过个人可标识信息（Personal Identifiable Information, PII）来聚合得到同一用户在不同网站的多个口令，将其作为用户口令库开展蜜口令库的安全性分析。尽管这种通过多个数据集聚合得到口令库的方式，能够得到更大规模的口令库数据，但 Cheng 等人的实验设定依然存在一系列问题。首先，Cheng 等人认为防御者拥有对测试集分布的全面了解，即防御者虽然在训练口令概率模型时不能提前获知用户口令库，但其训练数据和用户口令库来自相同的口令分布。然而，这一假设与现实场景存在巨大偏差，口令在口令语义、频率等各个层面的分布在不同网站、不同用户群体、不同时间上存在很大差别[68, 103]，防御者在训练口令概率模型时，不可能预先得到与目标用户群体口令分布完全一致的口令数据集。其次，Cheng 等人认为攻击者和防御者在信息上是对称的，二者在进行攻击、防御时所使用的口令数据集完全相同。这一假设同样不符合实际场景，按照 Kerckhoff[104, 105] 原则，攻击者可以获知防御者用于训练口令概率模型的完整数据集；此外，攻击者还可以进一步通过其它遭到泄露的口令数据集[9, 30, 106, 107] 来辅助进行攻击。如何设计与实际攻防场景更加贴近的蜜口令库安全性分析实验，依然是一个有待解决的重要问题。

4.1.2 本章贡献

围绕蜜口令库方案安全性分析和基于深度学习的蜜口令库方案，本章完成了以下工作：

- **基于量化变分自编码器对口令空间进行划分**。本文首先利用量化变分自编码器（VQ-VAE），从概率分布、语义特征两个方面出发，将口令空间划分为不同的子空间。利用 Transformer 模型作为先验概率模型、BERT 模型作为语义特征编码器，本文使用 VQ-VAE 将语义、分布上相似的口令映射

到高维空间中的相同离散向量上。此时，利用该向量对口令概率模型进行条件化，即让口令概率模型在离散向量诱导的条件概率空间里采样诱饵口令，实现对口令空间在语义、分布两个层面的划分。

- **深度学习自适应蜜口令库方案**。利用量化变分自编码器，本文成功对口令空间进行了概率分布和语义特征两个层面的切分，并且能够根据真实口令库中各个口令所处的子空间，针对性地进行编码和解码。这样一来，解码得到的诱饵口令，在分布和语义两个层面都和真实口令来自同一子空间，实现了自适应的诱饵口令库采样，且该自适应机制不会引入攻击者的区分攻击优势。同时，我们还根据蜜加密技术的编码解码场景，从一系列口令重用模型中进行筛选，利用 PointerGuess 作为重用机制的基础模型，并构造了基于条件概率空间的深度学习重用机制。
- **贴合现实攻防场景的安全性分析实验设计**。我们基于 30 个大规模真实口令数据集，在中英文两种语言的用户群体上，构造了三个不同的攻防场景。在选取防御者、攻击者的训练数据，以及用户口令库的测试数据时，我们从现实攻防场景出发，让攻击者不仅能够获得完整的防御者口令概率模型，而且还可以使用额外的、与用户口令库中服务类型、泄露时间更加贴近的口令数据集辅助区分攻击；同时，防御者和攻击者各自使用的口令数据集都与用户口令库存在差别，更加符合真实场景。实验结果表明，我们提出的蜜口令库方案在安全性上超越了现有的学术界各类方案，消融实验的结果则进一步说明了本章自适应机制和子空间切分方案的有效性。

第二节 相关工作

蜜口令库的安全性取决于生成的蜜口令库与用户真实口令库的不可区分性，即：

1. 诱饵口令库中的口令分布应当与用户的真实口令库尽可能相同，避免攻击者通过口令库的口令分布规律定位诱饵口令库。
2. 诱饵口令库应当能够重现用户的口令行为，如口令重用行为等，以避免攻击者通过观察口令库所表现出的用户口令行为特征来将真实口令库区分出来。

实现上述要点的关键挑战在于巧妙地设计自然语言编码器，特别是其中的口令概率模型（Password Probability Model, PPM）。在 IEEE S&P'15 中，Chatterjee 等[34] 采用 PCFG 模型作为口令概率模型，并且在自然语言编码器中引入子语法

机制，以重现用户的口令重用行为。上述方案被称为 NoCrack-NLE，是首个理论上可行的蜜口令库方案，但该方案的子语法机制存在歧义，因而会受到 Cheng 等[37] 提出的编码区分攻击。概括地说，编码攻击利用同一口令片段编码和解码规则不一致辨别真实口令库。例如，若 password 被固定为直接编码整个字符串，一旦被解密的种子未能解码出整个字符串（例如字符串 pass 和 word 的组合），则可判定解码口令库为假。同时，其选用的 PCFG 口令概率模型所建模的口令概率分布与真实口令库仍有较大差别，因而容易受到 Golla 等[35] 提出的 KL 散度区分攻击。

在 CCS'16 中，Golla 等[35] 首次提出了一种自适应机制，并基于 Markov 口令概率模型设计了其蜜口令库方案的自然语言编码器，即 Golla-NLE。该自适应机制从真实口令库的口令中随机选取一些 n-gram 字段，并提升这些字段在口令概率模型对应的概率，从而根据真实口令库适应性地生成诱饵口令库。相较于 NoCrack-NLE，上述自适应机制使诱饵口令库与真实口令库的口令分布更为相似，可以更好地抵抗基于真实口令库和诱饵口令库分布差异的 KL 散度攻击。然而，该自适应机制会泄露真实口令库中口令分布的信息，从而使其在面对提取真实口令分布的区分攻击[36] 下较为脆弱。

在 USENIX'19 中，Cheng 等[37] 提出的蜜口令库方案改进了 PCFG 口令概率模型，使其能够更好抵抗编码区分攻击。然而，该设计需要遍历所有的规则组合，导致指数级时间复杂度，在实际应用中面临着计算效率低下的问题。同时，在 USENIX'21，Cheng 等[36] 针对口令库中新添加的口令，进一步提出了增量更新机制，则可能导致解码出不完整的诱饵口令库，从而受到 Duan 等[38] 提出的特征区分攻击。

目前学术界现有的蜜口令库方案尽管各有亮点，并且各自能够规避某些区分攻击带来的危害，但都在面对其他区分攻击时较为脆弱，难以在实际应用中抵抗多样化的区分攻击。此外，时至今日，蜜口令库方案仍然停留在学术研究层面，而并未在工业界和实际场景中得到应用。如何设计一款能够较好抵御各类区分攻击的蜜口令库方案，并且将其投入到市场应用，在现实中保护用户口令等敏感信息，是一个尚未得到解决的问题。

第三节 预备知识

4.3.1 蜜加密技术与蜜口令库方案

蜜加密技术的基础如图 4.1 所示。给定一个消息空间 $\mathcal{M}$，其内部的各消息（总数有限）在消息空间中对应一个概率，蜜加密技术可以通过随机算法 encode，得到编码空间 $\mathcal{S}=\{0,1\}^l$ 上的一个编码，其中 l 是编码长度。相反，对于任意一个 $\mathcal{S}$ 上的编码，可以利用一个确定性算法 decode 来得到唯一对应的消息 m。

此时，构造一个编码空间，由于消息空间中消息总数有限，可以根据给定的规则对所有消息进行排序，并通过各消息的对应概率获得一个消息空间的累积分布函数。这样一来，对于任意一个消息，都可以确定其在累积分布函数中对应的位置与累积概率值，并且该概率值可以映射到编码空间的一个范围内。上述编码空间和消息空间的对应关系可以概括如下：

1. 给定一个消息空间中的消息，可以在编码空间中依概率取得多个与之对应的编码。
2. 给定一个编码空间中的编码，唯一确定一个消息空间中的消息。

建立上述关系以后，即可对一个给定的消息进行编码（得到多个可用编码时，均匀随机抽取其中一个作为编码），并且可以对该编码进行加密处理。解密时，若错误解密，得到的错误编码仍然在编码空间中唯一对应一个消息空间中的消息。这就使得攻击者在试图猜测密钥、进行解密时，难以通过是否可以获得正常解码结果（一般的加密在被错误解密时，得到乱码而非消息）来判断密钥是否正确。

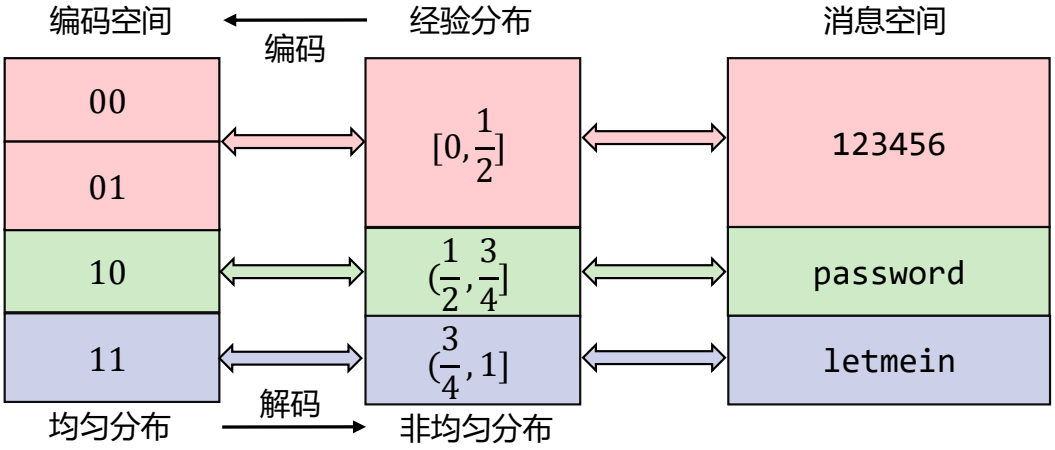


图 4.1 蜜加密技术基础：消息空间、概率分布和编码空间的对应关系

具体来说，对口令库明文的蜜加密，并不直接将口令作为消息，而是将口令解析为一系列口令规则，通过大量真实口令数据对这些规则的概率进行统计，进而将口令解析得到的规则作为消息进行蜜加密。这一过程则需要使用自然语言编码器（Natural Language Encoder）进行实现。本文在蜜加密的基础上，提出了全新的自然语言编码器和基于序列的口令概率模型、口令重用模型，以增强蜜口令库方案的安全性。

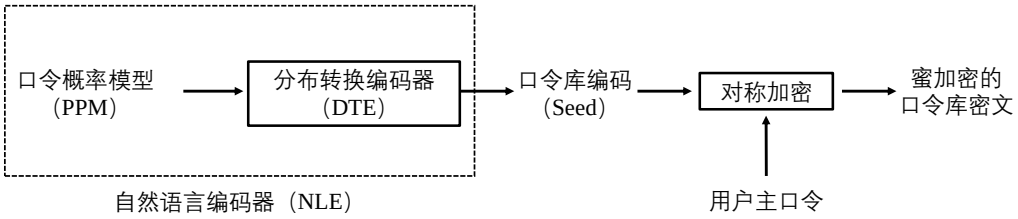


图 4.2 蜜口令库工作流程图

自然语言编码器（NLE）由口令概率模型（PPM）和分布转换编码器（DTE）组成，是设计一个安全的蜜口令库方案的关键。如图 4.2 所示，在蜜口令库中，用户的口令库中的每个口令首先会通过特定的 PPM 解析为有序生成规则序列和每个生成规则对应的概率值；然后，这些生成规则被 DTE 依次编码为二进制串（Seed），进而通过基于主口令的对称加密方案加密得到蜜加密的口令库密文。解密则相反，DTE 会将解密得到的二进制串序列依次解码为有序的生成规则序列，进而得到用户的口令库。

4.3.2 区分攻击

攻击者在对主口令进行离线猜测攻击时，将会解密得到一系列明文口令库。假设攻击者在进行 n 次猜测之后，能够成功猜中用户的主口令，那么攻击者需要从 n 个明文口令库中将用户的真实口令库区分出来。由于在线验证次数受限，攻击者的最优策略是按照一定的顺序，优先将最有可能是真实口令库的明文口令库进行在线验证。形式上，对于含有一个真实口令库和 $n-1$ 个诱饵口令库的一系列明文口令库 $\{v_1, v_2, \ldots, v_n\}$，攻击者使用评分函数 $s(\cdot)$ 对每个元素得到评分 $s(v_i)$。对所有评分进行降序排列后，真实口令库 v_{real} 的排名 r 定义为：

$$r = |\{v \mid s(v) > s(v_{real}), v \in \{v_1, v_2, \ldots, v_n\}, v \neq v_{real}\}| + 1$$

其中 $|\cdot|$ 表示集合中的元素个数。分数相同时，攻击者对所有分数相同的口

令库进行随机排列。那么这样一来，对于 k 个不同用户来说，蜜口令库方案的平均排名即为 $\bar{r} = \sum_{i=1}^{k} r_i/n$。此时，如果诱饵口令库和真实口令库不可区分，攻击者只能随机猜测，则平均排名为 0.5。该平均排名可以用于在安全性分析实验中评价蜜口令库方案的安全性，越接近 0.5 则蜜口令库方案越安全。

接下来我们来介绍学术界现有的区分攻击方案。由于本章主要分析现有蜜口令库方案中共有的脆弱性及其成因，所以此处不考虑只针对特定蜜口令库方案或 DTE 实例的区分攻击方案，如 Encoding 攻击[37, 38] 和 Adaptive 攻击[36]。

KL 散度攻击 该区分攻击方案由 Golla 等人在 2016 年提出[35]，用于攻击 NoCrack 蜜口令库方案。Golla 等人认为，NoCrack 所采用的 NLE 使用的是静态的口令概率模型，无法根据每个用户的口令库进行调整。只要用户口令库中的口令分布和 NLE 所描述的口令分布存在差别，就可以利用这种差别将真实口令库区分出来。KL 散度攻击所使用的评分函数如下：

$$\mathsf{s}(v) = \sum_{pw \in v} \frac{\mathrm{count}_v(pw)}{|v|} \lg\left(\frac{\mathrm{count}_v(pw)}{\mathrm{Pr}_{NLE}(pw)|v|}\right), \tag{4.1}$$

其中，$\mathrm{count}_v(pw)$ 是口令 pw 在口令库 v 内出现的次数，$|v|$ 是口令库 v 中的口令总数。$\mathrm{Pr}_{NLE}(pw)$ 是口令 pw 在蜜口令库方案的 NLE 下被分配的概率。实际上，该评分函数描述的是某一口令在口令库中出现的概率和在 NLE 中被采样得到的概率的差别。不难发现，采样自 NLE 的诱饵口令库的评分分数天然较低，因为其口令在口令库中的分布和在 NLE 中的分布在统计学上不存在显著差异。而只要用户口令的分布和 NLE 所描述的口令分布存在差别，该差别就可以被上述评分函数捕捉到，其评分会比诱饵口令更高，进而排名更加靠前。

单口令攻击 该区分攻击方案由 Cheng 等人在 2021 年提出[36]。在用户口令库中每个口令均为独立的假设下，单口令攻击的评分函数利用一个先验的口令概率分布（由辅助口令数据集得到），将口令库中每个口令在先验概率分布中的概率及其在 NLE 下的概率进行对比。形式上，该攻击的评分函数如下：

$$\begin{aligned} \mathsf{s}(v) &= \prod_{pw \in v} \mathrm{diff}(pw), \\ \mathrm{diff}(pw) &= \begin{cases} 1, & \textbf{If } f_D(pw) < 5 \textbf{ And } \frac{(f_D(pw)+1)/(|D|+1)}{\mathrm{Pr}_{NLE}(pw)} > 1 \\ \frac{(f_D(pw)+1)/(|D|+1)}{\mathrm{Pr}_{NLE}(pw)}, & \textbf{Otherwise} \end{cases} \end{aligned} \tag{4.2}$$

其中，diff(pw) 是利用辅助口令数据集来得到先验概率分布的启发式计算函数，$|D|$ 是辅助口令数据集里面所有口令的总数，$f_D(pw)$ 是口令 pw 在辅助口令数据集里出现的次数，$\mathrm{Pr}_{NLE}(pw)$ 则为蜜口令库方案所使用 NLE 下 pw 对应的概率（即被随机采样到的概率）。不难发现，单口令攻击的攻击的本质原理和 KL 散度攻击是一样的，即在用户口令和诱饵口令分别遵循不同分布的前提下，利用辅助数据集或者口令库中的口令频数，结合 NLE 所描述的诱饵口令分布，对这种分布上的差异进行捕捉和利用。

口令相似度攻击 这种区分攻击由 Cheng 等人在 2021 年提出[36]，其针对的不再是诱饵口令和真实口令在分布上的差异，而是口令库中所体现的口令重用行为。具体来说，口令重用行为可以用一系列相似度指标的数值进行描述，这些数值即为重用行为的特征。分别在真实口令库的数据集和一系列采样得到的诱饵口令库上对这些特征进行提取，即可通过贝叶斯分类器来根据口令库所体现的重用特征判断一个给定口令库是否为诱饵口令库。口令相似度攻击的评分函数如下：

$$\mathsf{s}(v) = \prod_{f \in \mathcal{S}_{\mathcal{F}}} \frac{\mathrm{Pr}_{real}(f = \mathbb{I}(v, vf))}{\mathrm{Pr}_{decoy}(f = \mathbb{I}(v, vf))} \tag{4.3}$$

其中，f 表示一个真值化的重用特征（拥有特征即为真，不拥有即为假），$\mathcal{S}_{\mathcal{F}}$ 是所有重用特征的集合；$\mathbb{I}(\cdot,\cdot)$ 为示性函数，如果 v 具有特征 f，则 $\mathbb{I}(v,f)$ 为真，否则为假。$\mathrm{Pr}_{real}(\cdot)$ 和 $\mathrm{Pr}_{decoy}(\cdot)$ 则分别为真实口令库、诱饵口令库上示性函数 $\mathbb{I}(\cdot,\cdot)$ 的概率分布。

上述形式化描述在直观上不便理解，此处举例加以说明。例如，假设有口令重用特征集合 $\mathcal{S}_{\mathcal{F}} = f_1, f_2$，在攻击者预先获取的一系列真实口令库中有 $\mathrm{Pr}_{real}(\mathbb{I}(v,f_1)) = p_1^{real}$ 和 $\mathrm{Pr}_{real}(\mathbb{I}(v,f_2)) = p_2^{real}$，也就是真实口令库中分别有 p_1^{real}、p_2^{real} 的口令库拥有 f_1、f_2 特征。而在攻击者预先获取的一系列诱饵口令库中，有 $\mathrm{Pr}_{decoy}(\mathbb{I}(v,f_1)) = p_1^{decoy}$ 和 $\mathrm{Pr}_{decoy}(\mathbb{I}(v,f_2)) = p_2^{decoy}$。那么，当一个口令库拥有 f_1 但不拥有 f_2 特征时，其评分函数的值为 $\frac{p_1^{real} * (1-p_2^{real})}{p_1^{decoy} * (1-p_2^{decoy})}$。

理论最优攻击 Duan 等人在 2024 年提出了基于蜜口令库方案和蜜加密技术中攻击者优势的区分攻击方案[38]。这种攻击方法利用用户主口令、口令库之间的关系，以及密文分布，对攻击者优势进行了一系列近似计算，得到了最终的评分函数。本文出于篇幅考虑，在此不对其理论推导进行介绍，直接介绍其最终的评分函数。口令库分布、主口令分布和密文分布分别为 $\mathrm{Pr}_{\mathrm{V}}(\cdot)$，$\mathrm{Pr}_{\mathrm{MP}}(\cdot)$，$\mathrm{Pr}_{\mathcal{C}}(\cdot)$。

同时假设 $\mathrm{Pr_V}(v) \approx Z \cdot \mathrm{Pr_{MP}}(mp)$（$Z$ 是常数），那么在忽略常数 $\frac{Z}{|\mathcal{S}|\mathrm{Pr_C}(c)}$ 的情况下，评分函数的表达式为：

$$\mathsf{s}(v) = \frac{\mathrm{Pr_V}(v)}{\mathrm{Pr_{NLE}}(v)} \cdot \mathrm{Pr_{MP}}(mp) \approx \frac{\mathrm{Pr_V}(v)^2}{\mathrm{Pr}_{NLE}(v)}. \tag{4.4}$$

其中 $\mathrm{Pr}_{NLE}(v)$ 表示 NLE 下口令库 v 的概率。在实际攻击过程中，攻击者可以使用口令概率模型、口令重用模型、口令数据集等方式来实例化 $\mathrm{Pr}_{NLE}(v)$。

4.3.3 文本分类模型

基于文本的语义特征来对文本进行分类，一直是自然语言处理领域的重要研究课题之一。近年来，一系列文本分类模型利用深度学习技术进行文本特征学习、感知和提取，在各类分类任务上达到了非常好的效果，实现了较高的准确率[108]。考虑到口令本身可以视为一种特殊的短文本，本章将利用这些先进的文本分类模型，构建基于语义对诱饵口令库和真实口令库进行区分的分类模型。

CNN 文本分类模型 2014 年，Kim 将卷积神经网络（CNN）应用于语句级的文本分类任务中，取得了较好的分类效果[109]。该模型通过对词嵌入进行一维卷积，进而提取并捕捉文本中的局部 n-gram 特征，进而在短文本分类任务中实现了较高的准确性。口令中的语义特征同样可以用 n-gram 进行较好捕捉[50]，所以利用 CNN 文本分类模型有望对诱饵口令和真实口令在语义特征上存在的微小区别进行有效区分，实现基于语义的区分攻击。

BERT 文本分类模型 2019 年，Devlin 等人[108] 利用 Transformer[67] 模型构建了 BERT（Bidirectional Encoder Representation from Transformers）模型，成功利用预训练技术增强了模型对自然语言特征的学习能力。BERT 模型能够通过掩码语言建模（Masked Language Modeling）在无标注的语料上进行预训练，同时学习自然语言中的双向特征，进而通过预训练方式直接从无标注数据中提取自然语言的深层次特征。在使用预训练后的 BERT 模型进行文本分类时，只需添加分类头并在分类任务训练集上进行模型微调，即可取得较为理想的分类效果。现有的一系列口令安全相关研究表明口令不仅具有双向性[22, 39, 50]，而且同样可以通过 BERT 的表征学习技术改进口令建模的准确性[14]，那么利用 BERT 对诱饵/真实口令进行分类同样是可行且有效的。

RoBERTa 文本分类模型 2021 年，Zhuang 等人[110] 在利用大量实验探索 BERT 模型有效性的基础上，提出 RoBERTa 训练方案对 BERT 模型进行改进。RoBERTa 对 BERT 模型的训练方式进行了健壮性优化，移除了 BERT 训练过程中的下一句预测任务，并且对掩码语言建模过程进行了动态调整，提升了模型训练过程的健壮性。同时，Zhuang 等人使用更大规模的训练数据进行了大规模训练，让模型能够在预训练过程中更充分地提取自然语言的特征。RoBERTa 后来成为了被广泛采用的 BERT 模型改进方案，将有可能在语义区分攻击中实现比 BERT 更高的攻击准确率。

DeBERTa 文本分类模型 2021 年，He 等人[111] 对 BERT 模型中的注意力机制和模型结构进行了修改，设计了改进后的 DeBERTa 模型。具体而言，DeBERTa 模型使用了解耦注意力机制（Disentangled Attention），其相对原始 BERT 中的注意力机制而言，删去了一些冗余的计算过程，能够更好捕捉文本中位置和内容的关系。此外，DeBERTa 还对掩码编码器进行了更改，使得掩码编码器能够在解耦注意力机制下捕捉到文本中不同部分的绝对位置。实验表明 DeBERTa 在一系列下游任务中的表现比 RoBERTa 和 BERT 更优，并且计算效率更高。

4.3.4 口令概率模型

能够对口令空间中的任意口令分配一个概率，并且口令空间中所有口令的概率之和为 1 的概率模型即为口令概率模型。一般来说，学术界所提出的口令猜测模型均为口令概率模型，其能够对训练集中的口令分布进行描述，进而按照概率降序生成口令猜测。本文将使用一系列口令猜测模型，将其作为口令概率模型来模拟攻击者所掌握的口令先验分布，并使用这些口令概率模型来实施基于口令分布的区分攻击。接下来本文将对学术界现有的一系列口令概率模型进行介绍。

PCFG/Markov 统计学口令概率模型[40, 50] 这两个模型都是学术界利用统计学模型设计的口令猜测模型。其中，PCFG 模型利用概率上下文无关文法（Probabilistic Context-Free Grammar）来统计口令在字段和字符两个层面的特征，而 Markov 模型则对口令数据集里的 n-gram 进行统计，进而捕捉口令的语义特征。这两个模型是较为经典的口令概率模型，被大量后续的学术界相关研究广泛使用。

RFGuess 经典机器学习口令概率模型[13] Wang 等人在 2023 年首次利用经典机器学习技术，使用随机森林（Random Forest）模型构建了 RFGuess 口令猜测模型。通过设计一系列启发式口令特征，Wang 等人成功利用随机森林的决策树机制和强大的特征捕捉能力，实现了非定向口令猜测攻击、基于个人可标识信息的定向口令猜测攻击，以及口令重用攻击。实验结果表明，RFGuess 在这三类攻击上的准确性达到了当时学术界各类口令猜测模型中的先进水平。

FLA 深度学习口令概率模型[39] Melicher 等人[39] 在 2016 年首次将深度学习技术引入到口令猜测领域，并基于该口令猜测模型设计了小体积、高准确性的非定向口令强度评估器。从当时已有的一系列口令猜测模型所能捕捉的口令特征出发，Melicher 等人在口令上下文信息、口令创建政策、口令逆序建模等一系列关键环节进行了较为科学的设计，实现基于 LSTM 深度学习模型的口令猜测模型。该模型在大猜测数下较为精确，超越了学术界此前设计的一系列统计学模型。这一口令猜测模型较快的推理速度、较小的体积占用、较为精确的口令分布建模，启发了后续的大量学术界相关研究，并因此得名（Fast, Lean, and Accurate, FLA）。

PassLLM 口令概率模型[41] 在 2025 年，Zou 等人利用基座大模型和微调技术，利用大模型强大的生成能力设计了 PassLLM 口令猜测模型。在使用合适的提示词和口令数据集对基座大模型进行微调之后，大模型即可快速准确地对口令概率进行计算，实现更准确的口令猜测攻击。Zou 等人所设计的 PassLLM 可以利用旧口令、PII 进行定向口令猜测攻击，也可以在没有额外信息的情况下实施非定向口令猜测攻击。本文主要使用其非定向口令猜测模型来实现对口令概率的建模，用于实施基于口令分布的区分攻击。

4.3.5 量化变分自编码器

量化变分自编码器（Vector-Quantised Variational-AutoEncoder, VQVAE）[112, 113] 是一种对连续特征进行离散化和隐式聚类的深度学习技术。VQVAE 的结构包括编码器、量化器和解码器，其中编码器从数据中学习连续的隐特征，并将输入数据映射到高维隐空间，量化器则利用编码得到的高维表征，以嵌入空间的形式学习一个码本（Codebook），该码本能够将高维隐空间中的任意一处映射到隐空间一个离散的向量上，实现了特征提取与隐式聚类的统一。解码器则在训练过程中通过数据映射得到的离散向量，尝试解码对原始数据进行还原，以此来

提升离散向量对数据特征的表达能力。

形式上，给定输入数据 $\mathbf{x}$，编码器提取特征 $\mathbf{z}_e = f_{enc}(\mathbf{x})$，量化器将该高维特征通过码本映射为 $\mathcal{C} = \{e_i\}_{i=1}^{K}$ 中的向量 $\mathbf{z}_q$。上述编码-映射-解码的学习任务，其损失函数由三部分组成，旨在最小化重构误差并同步编码器与码本：

$$\mathcal{L}_{VQ} = \underbrace{\mathbb{E}_{\mathbf{x}\sim p_{data}}\left[\|\mathbf{x} - g(\mathbf{z}_q)\|^2\right]}_{\text{Reconstruction}} + \underbrace{\mathbb{E}\left[\|\mathbf{z}_e - \text{sg}[\mathbf{z}_q]\|^2\right]}_{\text{Codebook Update}} + \underbrace{\beta\mathbb{E}\left[\|\text{sg}[\mathbf{z}_e] - \mathbf{z}_q\|^2\right]}_{\text{Commitment}}. \quad (4.5)$$

其中 sg[·] 表示停止梯度（stop-gradient）操作，β 为提交代价超参数，该设计促使相似特征被聚类到同一码字空间。实际上，训练完成之后，下游任务使用的往往只有编码器和码本，解码器则只在训练中通过重建损失来增强编码器和离散向量对数据特征的捕捉和建模能力。

4.3.6 自适应层归一化

根据某种给定的条件，对深度学习模型进行条件化，并指导模型的后续推理过程，是图像、文本生成领域较为热门的任务之一。自适应层归一化（Adaptive Layer Normalization, AdaLN）可以看作是对层归一化（Layer Normalization）的条件化扩展：它先对输入特征进行标准化，再由条件向量生成逐通道的缩放与平移参数，从而将外部信息注入到主干网络中[114–116]。

形式上，给定输入表示 $\mathbf{x}$ 和条件向量 $\mathbf{c}$，先对 $\mathbf{x}$ 做层归一化得到 $\text{LN}(\mathbf{x})$，再由条件映射函数生成调制参数 $\boldsymbol{\gamma}(\mathbf{c})$ 与 $\boldsymbol{\beta}(\mathbf{c})$，于是有

$$\text{AdaLN}(\mathbf{x}, \mathbf{c}) = \boldsymbol{\gamma}(\mathbf{c}) \odot \text{LN}(\mathbf{x}) + \boldsymbol{\beta}(\mathbf{c}). \quad (4.6)$$

其中 $\odot$ 表示逐元素乘法。在文本任务中，条件向量 $\mathbf{c}$ 可以来自类别标签、提示信息、长度统计量或其他先验特征，并被广播到序列中的各个位置；在图像生成中，这类条件化调制同样被广泛使用，尤其是在扩散 Transformer 等图像生成模型中，AdaLN 已成为常见的模型条件化方式之一[115, 116]。

4.3.7 口令数据集

本章所使用的真实口令数据集如表4.1所示（不包括口令库数据集 Pastebin）。在本章中，这些数据集将用于聚合得到用户口令库，并且用于蜜口令库方案的安全性分析。在清洗上述数据集时，本文首先清除掉所有长度大于 30 字符、小

表 4.1 本章所使用的真实口令数据集

语言	名称	服务种类	泄露时间	清除比例	剩余口令
Chinese	126	邮箱	2012	3.60%	6162669
	17173	游戏	2011	0.72%	18118140
	7k7k	游戏	2011	50.41%	7403497
	CSDN	技术论坛	2011	0.67%	6385450
	Dodonew	电子商务	2011	8.56%	13825474
	Ispeak	游戏	2011	1.72%	6780169
	Mop	游戏	N/A	1.73%	1856700
	Netease	邮件	2015	2.55%	253889121
	Renren	论坛	2011	7.13%	4368406
	Sina	社交媒体	2020	2.22%	18953297
	Sohu	论坛	2015	5.08%	13894741
	Taobao	电子商务	2012	1.28%	14878848
	Tianya	论坛	2011	1.88%	30606549
	Youku	视频	2016	28.05%	66583261
	Ys168	分享平台	N/A	3.63%	316594
English	Brazzers	成人网站	2013	0.97%	919045
	Clixsense	调研服务	2016	≤0.01%	2221855
	Couponmom	优惠服务	2014	2.72%	10628043
	Dailyquiz	教育	2021	11.65%	11366790
	Datpiff	音乐	2021	3.49%	6932601
	Duelingnetwork	游戏	2017	4.65%	5901716
	Imgur	社交媒体	2013	0.02%	1755093
	Ixigo	旅游	2019	1.03%	3538983
	Lookbook	社交媒体	2012	9.17%	971349
	Mate1	网上约会	2016	10.26%	24590838
	Mathway	教育	2020	2.19%	15701738
	Twitter	社交媒体	2022	4.34%	24465322
	Warmane	游戏	2016	2.05%	484272
	Yahoo	门户网站	2013	4.24%	434209
	Yahoovoice	语音通信	2012	4.46%	248205

* 7k7k 数据集未使用任何口令长度限制，导致有大量口令长度小于 6。
* Youku 数据集存在大量”null” 的口令，在数据清洗中均被排除。

于 6 字符的口令；同时，由于后续的口令库聚合需要使用用户的邮箱地址，所有邮箱地址明显不属于真实邮箱地址的对应口令也被排除在实验之外（例如所有不包含”@” 字符的邮箱地址，所对应的口令均被排除）。

第四节 基于子空间划分的自适应蜜口令库方案

在蜜口令库方案攻防场景中，防御者首先需要将口令概率模型在训练口令数据集上进行拟合，构建能够描述该数据集口令分布的口令概率模型，进而构建 NLE。随后，在加密过程中，对于一系列用户的口令库，防御者通过上述 NLE，在蜜加密技术下，利用每个用户设定的主口令对他们的明文口令库进行加密，分别得到每个用户各自的口令库密文。攻击者则需要在防御者完成上述过程后，通过各种手段获取每个用户的口令库密文，并且尝试对其进行主口令猜测，对每个口令库密文解密得到一系列明文口令库。

假设攻击者能在进行一定次数的主口令猜测后，正确猜中用户主口令，那么攻击者的目标是，通过区分攻击中的评分函数，对解密得到的大量明文口令库进行排序，尽可能降低用户真实口令库在所有明文口令库中的排序。

在现实攻防场景中，根据 Kerckhoff 原则，攻击者可以获得完整的 NLE 及其口令概率模型，并且可以通过随机采样从 NLE 中得到大量的诱饵口令，对这些诱饵口令中的特征进行提取。此外，考虑到攻击者有后手优势，攻击者还可以使用辅助口令数据集，尽可能去对真实用户的口令分布进行模拟和逼近，实现比防御者更精确的口令分布建模。具体来说，攻击者可以从口令数据集中提取得到口令库特征集合 $X=\{f_1,f_2,\ldots,f_n\}$，并且分别在采样得到的诱饵口令数据集、辅助口令数据集里分别提取诱饵口令库、真实口令库具有的特征。这时，给定一个具有特征 $x\in\{f_1,f_2,\ldots,f_n\}$ 的明文口令库，其为真实口令库的后验概率为：

$$\Pr(\text{Real}\mid x)=\frac{\Pr(\text{Real})\Pr(x\mid\text{Real})}{\Pr(\text{Real})\Pr(x\mid\text{Real})+\Pr(\text{Decoy})\Pr(x\mid\text{Decoy})} \tag{4.7}$$

考虑到在区分攻击中，攻击者不关心上述后验概率的具体值，只需要根据其相对值对不同的明文口令库进行评分和排序，那么评分函数可以在后验概率的基础上改写为：

$$s(x)=\frac{\Pr(x\mid Real)}{\Pr(x\mid Decoy)} \tag{4.8}$$

不难得出，利用 $s(x)$ 和 $Pr(Real\mid x)$ 对明文口令库进行排序，得到的排序结果是相同的。从评分函数 $s(x)=\frac{\Pr(x|Real)}{\Pr(x|Decoy)}$ 可以看出，攻击者的区分攻击优势来自于真实口令库和诱饵口令库在特征 x 上的分布差异。

对于防御者而言，若要提升蜜口令库方案的安全性（即诱饵口令库、真实口令库在利用上述评分函数的区分攻击中，平均排序尽可能接近 0.5），就需要使 $\Pr(x\mid Real)$ 和 $\Pr(x\mid Decoy)$ 这两个条件分布尽可能接近。这时，由于在编码过程中防御者可以获取单个用户的真实口令库明文，那么此时可以利用真实口令库中的信息对 NLE 进行适应性调整，使得采样得到的诱饵口令在特征分布上向真实口令库靠拢，从而缩小 $\Pr(x\mid Real)$ 和 $\Pr(x\mid Decoy)$ 之间的差距。这一自适应过程实际上公开了部分真口令库的相关信息，攻击者可以通过该信息，辅助其提升区分攻击的成功率。

为了避免自适应机制所公开的信息为攻击者提供额外的区分攻击优势，本文提出基于子空间划分的自适应蜜口令库方案。该方案的核心思想是，将口

令库空间 $\mathcal{P}$ 按照某种特征划分为若干互不相交的子空间 $\mathcal{P}_1,\mathcal{P}_2,\ldots,\mathcal{P}_K$，其中 $\mathcal{P}=\bigcup_{k=1}^{K}\mathcal{P}_k$。注意子空间划分的过程不改变特征在整个口令空间 $\mathcal{P}$ 上的总体分布。在编码时，防御者只在真实口令库所属的子空间对其按照蜜加密机制进行编码。然后，在解码时，防御者公开真实口令库所属的子空间索引 k，且仅在该子空间 $\mathcal{P}_k$ 内部进行解码。此时，攻击者所观察到的自适应调整行为对所有落入 $\mathcal{P}_k$ 的口令库都是一致的。因此，攻击者无法仅从自适应机制所公开的信息对真实、诱饵口令库进行区分。与此同时，在同一子空间 $\mathcal{P}_k$ 内部，诱饵口令和真实口令在特征分布上更加接近，缩小 $\Pr(x\mid Real,\mathcal{P}_k)$ 和 $\Pr(x\mid Decoy,\mathcal{P}_k)$，安全性可以得到进一步提升。

本文后续将分别从口令概率分布和口令语义特征两个方面出发，利用 VQ-VAE 对口令空间进行划分，并基于此构建自适应蜜口令库方案。

第五节 基于 VQ-VAE 的自适应口令概率模型

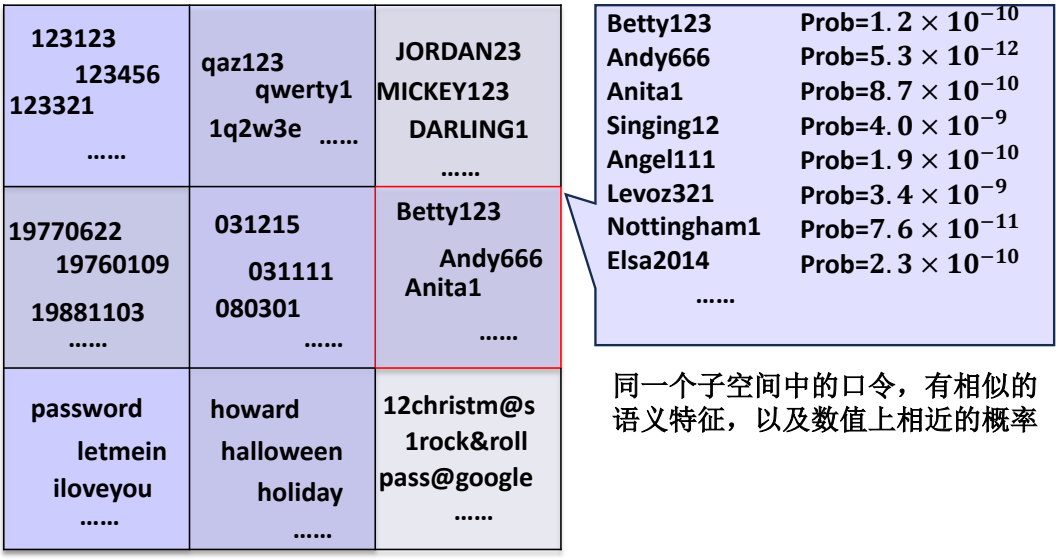


图 4.3 从语义、分布两类特征出发的子空间切分机制样例。

下面本节将对本研究设计的自适应口令概率模型进行介绍。从宏观上，如图4.3所示，该自适应口令概率模型能够从口令概率分布、口令语义特征两个方面，分别对口令空间进行切分，得到离散的子空间。通过公开真实口令分别所属的概率分布、语义特征两类子空间，本研究设计的深度学习蜜口令库方案将在对应的子空间上对用户口令进行编码。这样一来，攻击者在使用错误主口令解密时，将得到与真实口令处于同一空间的诱饵口令，让诱饵口令在分布、语义上与真实口令更相似，更加难以区分。

4.5.1 利用 VQ-VAE 进行口令空间切分

VQ-VAE 能够通过编码器学习到的数据高维表征，对任意数据通过其在高维空间中的表征，映射到量化器码本中的一个离散向量上，实现了对高维空间的切分。接下来我们将介绍如何使用 VQ-VAE 对口令空间进行切分。

对于一个已经在口令数据集上训练过的 VQ-VAE，设口令空间为 $\mathcal{P}$，编码器为 $f_{enc}:\mathcal{P}\to\mathbb{R}^d$，量化器对应的码本为 $\mathcal{E}=\{\mathbf{e}_k\}_{k=1}^K\subset\mathbb{R}^d$。对于任意口令 $pw\in\mathcal{P}$，编码器先将其映射为高维表征 $\mathbf{z}_e=f_{enc}(p)$。随后，量化器通过最近邻规则选择最接近的码字：

$$q(\mathbf{z}_e)=\mathbf{e}_{k^*},\qquad k^*=\arg\min_{1\le k\le K}\|\mathbf{z}_e-\mathbf{e}_k\|_2^2, \tag{4.9}$$

并将 p 映射到离散隐向量 $\mathbf{e}_{k^*}$。因此，在固定编码器和码本参数后，VQ-VAE 实际上诱导出一个从口令空间到码本索引集合的划分：

$$\mathcal{P}=\bigcup_{k=1}^{K}\mathcal{P}_k,\qquad \mathcal{P}_k=\{p\in\mathcal{P}\mid q(f_{enc}(p))=\mathbf{e}_k\}. \tag{4.10}$$

这样一来，编码器为每个口令给出唯一的连续表征，量化器再按最近邻规则将该表征投影到码本上的一个离散向量上，从而诱导出对口令空间的划分。同一子空间内的口令在编码器所定义的高维表征上通常更接近，子空间之间的边界则由码本中各离散向量在隐空间中的位置决定。

不难看出，上述由 VQ-VAE 进行的子空间划分，是以编码器学习得到的口令表征作为依据的。接下来，本文将分别以口令概率分布和语义特征两个口令特征出发，对上述 VQ-VAE 的子空间切分方案进行实例化。

4.5.2 口令概率分布的 VQ-VAE 空间切分

利用口令概率分布作为口令特征，并进行基于 VQ-VAE 的子空间切分，需要面临两个主要挑战。首先，使用什么口令概率模型对概率分布进行建模，需要有准确性、模型体积与推理速度两个方面的考量；随后，该如何利用口令概率模型所描述的经验口令分布，设计 VQ-VAE，同样需要在技术上加以推敲。

基于小体积 Transformer 解码器的先验口令概率建模 Transformer 模型[67] 以其强大的语言建模能力，被大量自然语言处理的相关研究广泛使用，口令安全领域的近期研究也展现了 Transformer 模型在口令建模上的优越性[14, 58]。Cheng 等

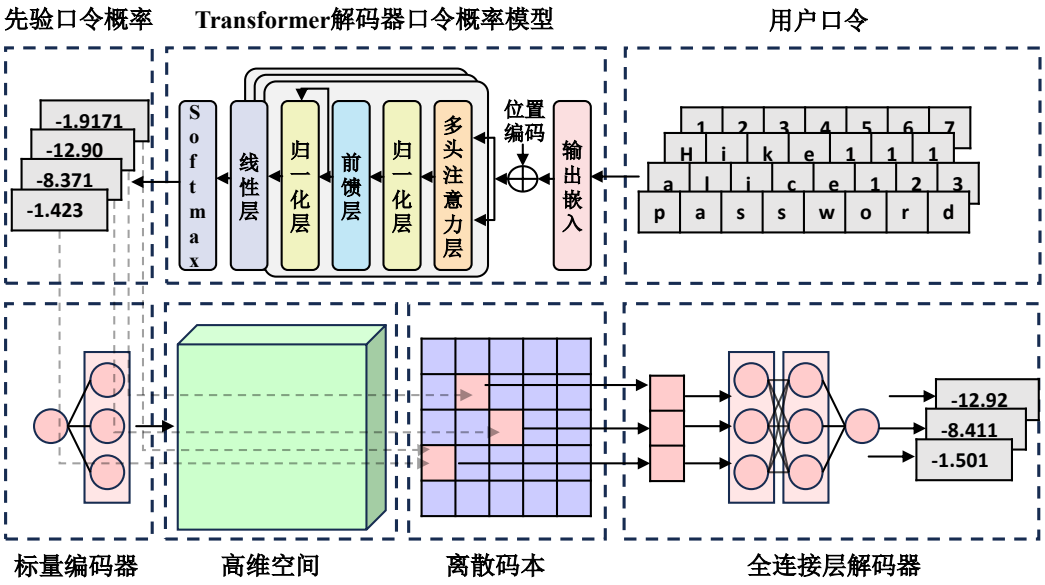


图 4.4 利用 VQ-VAE，基于口令概率分布对口令空间进行切分。

人在 CCS'25 中使用 Transformer 解码器对口令库进行建模，直接使用该模型作为口令概率模型，实现了比基于统计学模型的蜜口令库方案更高的安全性[102]。基于上述研究中 Transformer 模型所展现出的强大语言建模能力，特别是在口令概率分布和口令库建模上的优越性，本研究将利用 Transformer 模型作为建模先验口令概率的基础模型。

在蜜口令库方案中使用 Transformer 模型的一大挑战是其体积过大。为此，由于此时只需要利用 Transformer 对口令概率进行初步建模，不需要其具有生成能力，本研究在口令概率分布的 VQ-VAE 中只使用小参数的 Transformer 解码器作为先验口令概率模型。该口令概率模型在单个口令组成的口令数据集上进行训练之后，其描述的口令概率分布，就是本研究在口令概率分布层面上对口令空间进行切分的先验口令概率。该模型为口令逐字符分配的概率，将作为口令在概率分布上的表征，由 VQ-VAE 的编码器和量化器进行建模并量化，最终实现口令空间在概率分布上的切分。

以字符累乘概率为 VQ-VAE 编码器特征 口令概率同样有多种表征设计路径。一方面，可以依赖口令概率模型对口令逐字符分配的概率，将口令中每个字符概率组成的列表作为口令概率分布的表征。另一方面，也可以对每个字符的概率进行累乘，得到一个标量累乘概率，直接将其作为概率分布表征输入到 VQ-VAE 中进行编码和量化。

本研究根据蜜口令方案的实际场景和需求，选择利用字符的累乘概率作为

口令在概率分布上的表征，并利用 VQ-VAE 对该表征进行离散化和口令空间切分。在蜜口令库方案中，利用口令概率分布对口令空间进行切分的主要目的是让解码得到的诱饵口令在概率上尽可能和真实口令相对一致，进而抵御基于分布的区分攻击。由于攻击者的分布感知攻击方案在评分函数中直接采用整个口令的概率对口令库进行记分，按照口令在蜜口令库方案中的编码解码方式，使用口令逐字符累乘概率作为口令概率进行空间切分[35, 36, 49]，才可以达到上述目的。

相反，使用口令逐字符概率序列作为概率分布表征，容易让 VQ-VAE 按照口令的语义特征，特别是首个字符的语义特征来对口令空间进行切分。这是因为在字符级口令概率模型中，不论口令后续字符是什么，口令第一个字符的概率是固定的。例如，*abc123* 和 *aaa111* 中，第一个字符 *a* 的概率是相同的，但二者后续的逐字符概率则完全不同。此时，如果将概率序列作为表征，模型很容易学习到首字符概率相对固定这一特征，进而根据首字符概率对口令空间进行划分，无法达到让诱饵口令和真实口令在概率上贴近的目的。

使用全连接层和线性层设计编码器、解码器 在设计先验口令概率模型和口令概率分布表征之后，需要进一步设计编码器和解码器，让编码器能够将概率相近的口令映射到相同的离散向量中（即相同的子空间）。考虑到本研究所采用的概率表征实际上是标量，其信息量较小，故只需要使用小体积的线性层即可将标量概率值映射到高维空间。同样地，由于在训练离散码本和编码器时只需让二者学习概率密度的分布，训练过程中解码器使用简单的全连接层，根据编码器和量化器给定的离散向量输出单一的标量。在实际训练过程中，处于 $[0,1]$ 范围内的概率值两两之间数值差异过小，不易于线性层和量化器学习并映射，所以本研究采用 $log_{10}(\mathrm{Pr}_{prior}(pw))$ 作为编码器输入，以放大不同口令先验概率在数值上的差异。

总结。通过小体积 Transformer 解码器作为口令概率模型，并利用线性层作为标量的编码、解码器，即可使用 VQ-VAE 对口令按照其在先验口令分布中的概率值，将其映射到一个高维空间的离散向量上。在 VQ-VAE 训练完成之后，此处的解码器可以直接丢弃，只使用编码器和 Transformer 口令概率模型作为后续的 NLE 组件。

4.5.3 口令语义特征的 VQ-VAE 空间切分

接下来本文介绍如何对口令语义特征进行建模与提取，并利用 VQ-VAE 将语义特征上相似的口令映射到相同的离散向量上。

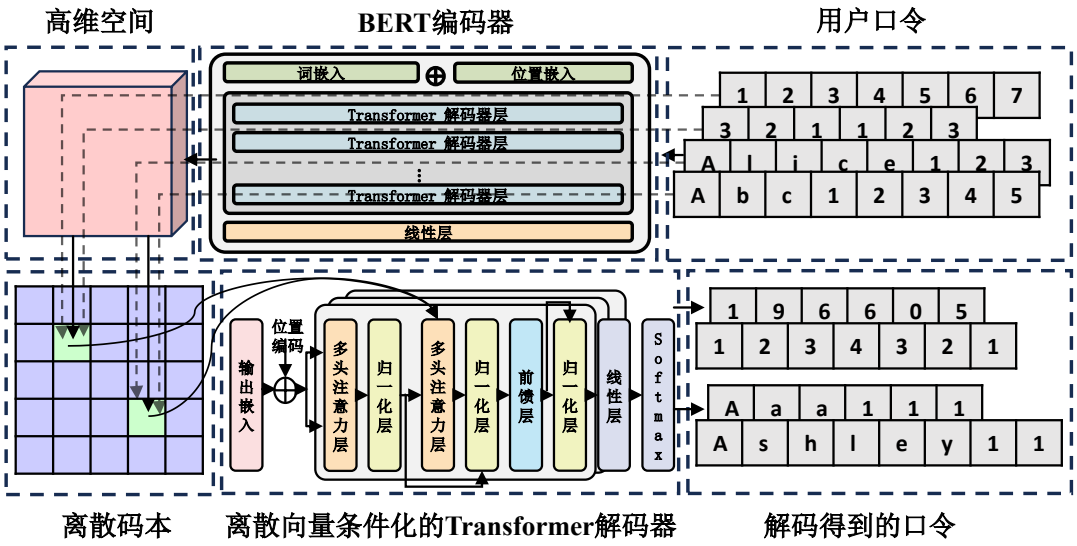


图 4.5 利用 BERT 和 Transformer 模型构建 VQ-VAE，基于口令语义特征对口令空间进行切分。

利用 BERT 编码器提取语义表征 现有研究指出口令具有一定的双向性[14, 39, 50, 58]，并且预训练后的口令模型可以较好捕捉口令语义特征[14]。因此，我们采用小体积的 BERT 模型[108] 作为编码器，从无标注的口令数据中提取口令语义特征，并通过 VQ-VAE 对 BERT 中的语义特征进行聚类，进而实现基于口令语义特征的口令空间划分。

首先，本研究不直接使用在大规模自然语言数据集上预训练的 BERT 模型，而是从头自行构建未经预训练的 BERT 模型，从字符层面对口令语义特征进行建模。该设定考虑了模型体积和实际场景的需求：如果使用在自然语言上预训练的 BERT 模型，其词表较大，模型体积同样较大，不利于作为蜜口令库方案的组件在用户端部署；口令本身是短文本，虽然从单词层面出发、利用自然语言数据集进行预训练有助于提升表征建模能力，但从字符层面建模有利于更细粒度、小尺度的微观语义建模[14]。在构建小体积字符级 BERT 模型之后，本研究将其在口令数据集上进行掩码语言建模的预训练，让其在无标注口令数据集上学到丰富的口令语义特征。

随后，在 VQ-VAE 的训练中，本研究冻结 BERT 模型的权重，将其作为编码器对口令进行编码，其池化层的输出作为量化器的输入，并由量化器对其语义特征进行量化，映射到码本中的离散向量上。VQ-VAE 在训练过程中以 Transformer 模型作为解码器，下面我们对该过程进行具体介绍。

使用 Transformer 模型作为解码器 BERT 编码器在预训练过程中对口令语义特征进行建模，VQ-VAE 则在训练过程中对这些语义特征进行映射，将语义上相似的口令映射到同一离散向量上。实现这一映射、聚类的关键在于构建合适的解码器，其能够对量化器码本中各离散向量代表的语义特征进行建模，尝试通过离散向量反向重构原始口令，进而对量化器的映射、聚类效果进行评判和指导。

本研究对 Transformer 模型架构稍加改造，将改造后的模型作为解码器实现上述重建过程。在量化器将口令根据语义特征映射到一个离散向量 z_q 之后，解码器首先将该向量通过线性层投影到 Transformer 模型的 Memory 中，让该离散向量对 Transformer 解码器进行条件化。随后，由 z_q 条件化的 Transformer 模型根据投影得到的 Memory 对原始口令进行重建。在 VQ-VAE 的训练过程中，该具有生成能力的 Transformer 通过 BERT 模型编码得到的 z_q 尝试对口令进行重建，进而反过来对 z_q 所能描述的语义特征进行质量上的监督，让量化器对语义上相似的口令映射到相同的离散向量上。

4.5.4 自适应口令概率模型

现在本文介绍如何根据上述口令概率分布、语义特征两个不同方面的子空间切分方案，实现自适应口令概率模型。

AdaLN 双路条件化的 Transformer 解码器 在上一小节里，本文说明了如何使用 VQ-VAE 将口令通过编码器和量化器，以离散向量的形式，基于分布和语义两类特征映射到不同的子空间中。上述过程中的口令子空间是由高维空间中的离散向量诱导的，并不是对口令空间的直接划分。所以，如何根据离散向量条件化口令概率模型，使得模型能够在子空间的限制下采样诱饵口令、编码真实口令，成为了本研究需要解决的又一个技术难题。

本研究设计了基于自适应层归一化（Adaptive Layernorm, AdaLN）的条件化 Transformer 解码器层（简称 AdaLN-Transformer 解码器），实现了在分布、语义特征离散向量约束下的子空间诱饵口令采样和真实口令编码。AdaLN-Transformer 解码器的基本结构如图4.6所示，主要组成部分包括双路 AdaLN 参数调制器和包含 AdaLN 调制模块的 Transformer 解码器。其中，AdaLN 参数调制器以语义特征离散向量 z_{bert} 和分布特征离散向量 z_{prob} 作为输入，并为 Transformer 解码器输出两组尺度/偏移变换参数 (s_{prob}, h_{prob}) 和 (s_{bert}, h_{bert})。对于批次大小为 B 的一系列口令，分布、语义编码器分别编码得到 $z_{bert} \in \mathbb{R}^{B \times d_{bert}}$ 和 $z_{prob} \in \mathbb{R}^{B \times d_{prob}}$，

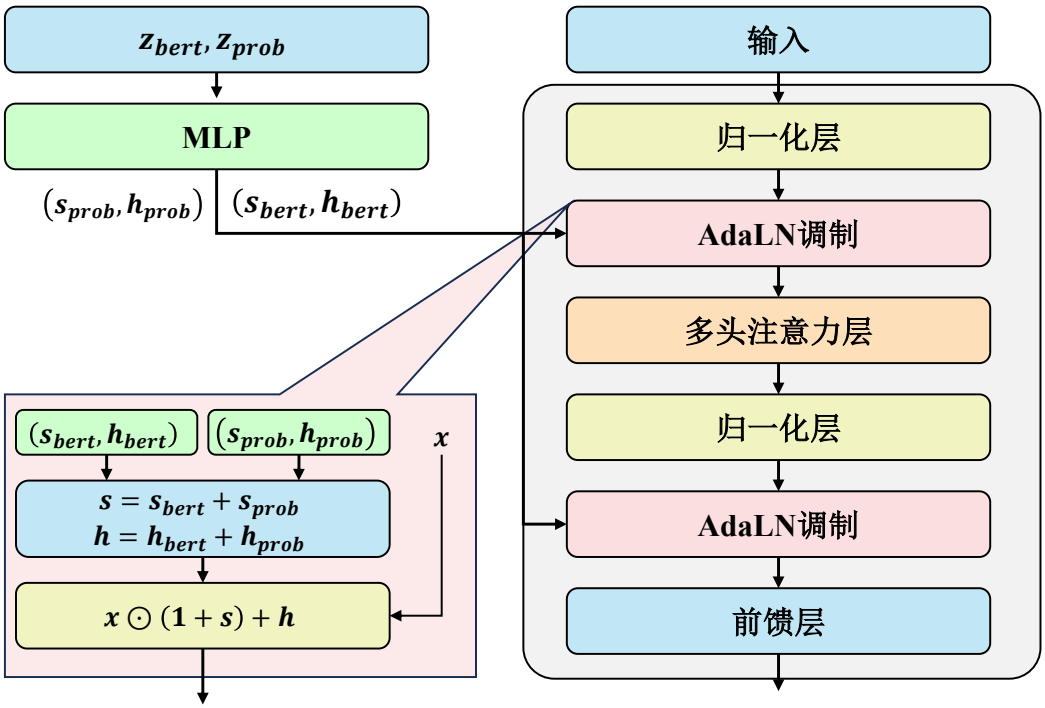


图 4.6 基于 AdaLN 的条件化 Transformer 解码器层。

AdaLN 参数调制器首先将 $z_{bert} \in \mathbb{R}^{B \times d_{bert}}$ 和 $z_{prob} \in \mathbb{R}^{B \times d_{prob}}$ 通过两个独立的 MLP 映射到 Transformer 模型对应的高维空间里:

$$u_{prob} = MLP_{prob}(z_{prob}) \in \mathbb{R}^{B \times 2NH}, \qquad u_{bert} = MLP_{bert}(z_{bert}) \in \mathbb{R}^{B \times 2NH}. \tag{4.11}$$

其中，N 为 Transformer 解码器层数，H 每层的隐藏层维度。将 u_{prob} 按最后一维分组，对于第 l 层，其变换参数为：

$$\begin{aligned} s_{prob}^{(l)} &= u_{prob}[:, 2lH : (2l+1)H], \\ h_{prob}^{(l)} &= u_{prob}[:, (2l+1)H : (2l+2)H], \end{aligned} \qquad l = 0, \ldots, N-1. \tag{4.12}$$

从 u_{bert} 中用同样方法可得 $s_{bert}^{(l)}, h_{bert}^{(l)}$。其中 $s_{prob}^{(l)}, h_{prob}^{(l)}, s_{bert}^{(l)}, h_{bert}^{(l)} \in \mathbb{R}^{B \times H}$，分别表示该层的通道尺度和偏移变换参数。

随后，即可在 Transformer 解码器中，利用上述变换参数，在多头注意力层和前馈层之前，分别对模型推理过程进行条件化的调整，实现 z_{prob} 和 z_{bert} 条件化的口令建模。此处以解码器第 l 层里，多头注意力层之前、归一化后的推理结果 $X \in R^{L \times B \times H}$ 为例说明如何使用变换参数对模型进行条件化调制。首先将先前的 $(s_{prob}^{(l)}, h_{prob}^{(l)})$ 和 $(s_{bert}^{(l)}, h_{bert}^{(l)})$ 在长度上广播，并且加和：

$$s^{(l)} = s^{(l)}_{prob} + s^{(l)}_{bert}, \qquad h^{(l)} = h^{(l)}_{prob} + h^{(l)}_{bert}. \tag{4.13}$$

经过上述参数对 X 进行条件化调制：

$$X' = X \odot (1 + s^{(l)}) + h^{(l)}, \tag{4.14}$$

其中 $\odot$ 表示逐元素乘法。在后续的多头注意力机制里，z_{prob} 和 z_{bert} 条件化调制后的 X' 将替代 X 作为输入。在前馈层之前进行 AdaLN 的计算方式与上述过程相同，此处不再重述。

多尺度特征条件化的自适应口令概率模型 使用 AdaLN 对 Transformer 解码器进行离散向量的条件化调制，可以在一定程度上将解码器对口令的建模限制在给定的口令子空间里（由分布、语义两类离散向量诱导）。为了更好让自适应口令概率模型生成在分布上和真实口令一致的诱饵口令，本研究在 AdaLN-Transformer 解码器的基础上，添加了口令分布特征融合机制。如图4.7所示，对于先前各个时间步的模型输出（即已经生成了部分字符的口令），将其通过口令分布特征融合机制，与历史输出的各时间步字符先验概率、真实口令对应的离散向量 z_{prob} 结合起来，共同作为 AdaLN-Transformer 的输入。

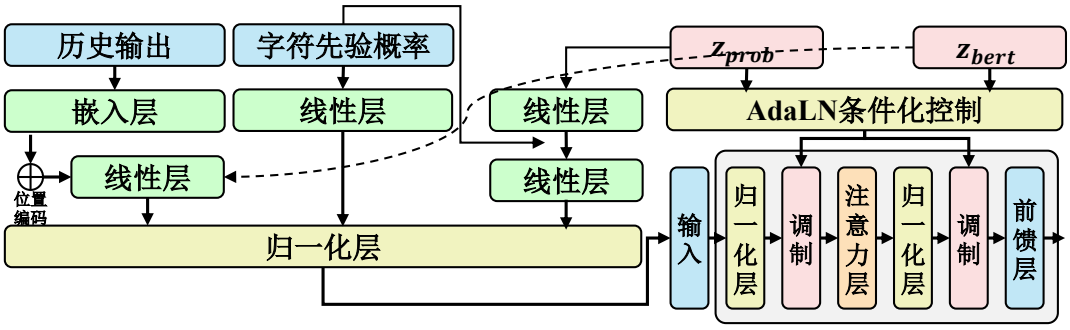


图 4.7 概率-语义双条件化的口令概率模型。

具体来说，给定本研究在4.5.2中提到的 Transformer 先验口令概率模型 LM，其在位置 i 处给出的下一字符条件对数概率向量为：

$$\boldsymbol{\pi}_i = \log \Pr_{\mathrm{LM}}(\cdot \mid c_{<i}) \in \mathbb{R}^{|\mathcal{V}|} \tag{4.15}$$

其中 $|\mathcal{V}|$ 为字符集大小。对一段已解码的口令 $c_0 c_1 \ldots c_{L-1}$（c_0 为起始符），其在 LM 下的逐字符对数概率为：

$$\log \Pr(c_i \mid c_{<i}) = \boldsymbol{\pi}_i[c_i], \qquad i = 0, 1, \ldots, L-1, \tag{4.16}$$

$[\cdot]$ 表示取向量中对应下标的分量，并规定 $\log \Pr(c_0 \mid c_{<0}) = 0$。前缀 $pw_t = c_0 c_1 \ldots c_t$ 的逐字符累积先验对数概率为：

$$\log \Pr_{\mathrm{LM}}(pw_t) = \sum_{i=0}^{t} \log \Pr(c_i \mid c_{<i}). \tag{4.17}$$

分布离散向量 $\mathbf{z}_{\mathrm{prob}}$ 经可学习线性映射预测一个全局预算目标：

$$s = f_s(\mathbf{z}_{\mathrm{prob}}) \in \mathbb{R}, \qquad f_s : \mathbb{R}^{d_p} \to \mathbb{R}. \tag{4.18}$$

则位置 t 处、由 $\mathbf{z}_{\mathrm{prob}}$ 诱导的剩余预算为 $s - \log \Pr_{\mathrm{LM}}(pw_t)$。对每个位置 t，构造三元标量特征：

$$\mathbf{f}_t = \left[\log \Pr(c_t \mid c_{<t}),\ \log \Pr_{\mathrm{LM}}(pw_t),\ s - \log \Pr_{\mathrm{LM}}(pw_t) \right]^{\top} \in \mathbb{R}^3. \tag{4.19}$$

以上计算对口令各个位置上的字符并行完成，得到特征矩阵 $\mathbf{F} = [\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_{L-1}] \in \mathbb{R}^{L \times 3}$。这时，上述特征包含了当前已有口令字符在先验口令概率模型中的逐字符概率，同时还包括 z_{prob} 诱导的口令子空间中各所对应的口令概率范围。模型可以通过上述信息，感知每个字符在先验概率分布中对应的概率，并且合理按照 z_{prob} 选择下一时间步选择带有何种先验概率的字符、决定何时终止口令序列生成（即在下一时间步选择序列终止符）。这样一来，从 z_{prob} 条件化的口令概率模型里所采样得到的诱饵口令，就会在先验口令概率上与原始口令保持相对一致。

由于条件化口令概率模型还需要根据 z_{bert} 调整所生成口令的语义特征，此时需要继续构建后续的融合特征输入。已生成口令的字符嵌入经位置编码与语义条件注入后，得到：

$$\tilde{\mathbf{e}}_t = \mathrm{Embed}(c_t) + \mathrm{PE}(t) + f_{\mathrm{bert}}(\mathbf{z}_{\mathrm{bert}}), \tag{4.20}$$

其中 $g_{\mathrm{bert}} : \mathbb{R}^{d_b} \to \mathbb{R}^H$ 为线性层。将 LM 在当前位置的对数概率向量 $\boldsymbol{\pi}_t$ 与标量特征 $\mathbf{f}_t$ 分别投影后拼接，经融合线性层得到 AdaLN-Transformer 的初始隐状态：

$$\mathbf{h}_t^{(0)} = \mathbf{W}_{\mathrm{fuse}} \left[\tilde{\mathbf{e}}_t \parallel \mathbf{W}_{lm} \boldsymbol{\pi}_t \parallel \mathrm{MLP}_s(\mathbf{f}_t) \right] \in \mathbb{R}^H, \tag{4.21}$$

其中 $\|$ 为向量拼接操作，$\mathbf{W}_{lm} \in \mathbb{R}^{(H/2)\times|\mathcal{V}|}$，$\mathbf{W}_{\text{fuse}} \in \mathbb{R}^{H\times(H+H/2+H/4)}$。通过上述过程得到的模型输入，可以综合 z_{prob}、z_{bert} 以及当前已生成口令字符的先验概率、累积概率等特征，让模型在 AdaLN 的基础上，根据已经生成的当前口令字符，更好地对下一时间步应该选择何种概率、何种语义的字符进行建模。

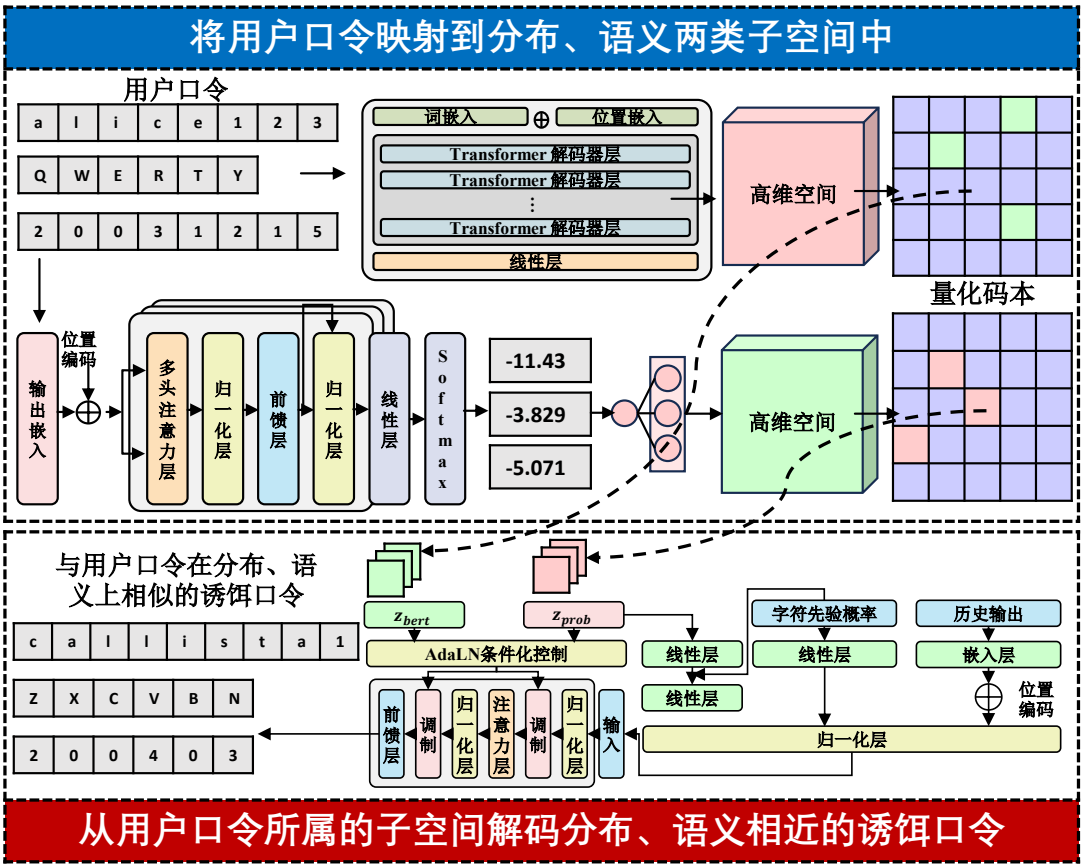


图 4.8 利用 VQ 将用户口令映射到离散的子空间中，并在对应子空间里采样得到与用户口令在语义、分布上接近的诱饵口令。

VQ 自适应的 NLE 工作流程 现在，本文介绍整个自适应口令概率模型在对单个用户口令进行编码，并根据用户口令分布、语义特征采样诱饵口令时的工作流程。如图4.8所示，在编码过程里，对于给定的用户口令，使用4.5.2和4.5.3中描述的分布、语义编码器和量化器，将用户口令映射到分布、语义两类口令子空间中。随后，公开用户真实口令所属的子空间，在对单个口令进行编码、解码时，用公开的两类子空间对自适应口令概率模型进行条件化。此时，从自适应口令概率模型里采样得到的口令，其所属的分布、语义两类子空间将会和用户口令相同，进而在分布、语义特征上与用户口令相似，实现自适应的口令编码、解码。

第六节 基于 PointerGuess 的口令重用机制

4.6.1 口令重用模型的筛选标准

用户的口令重用行为非常复杂，涉及对口令在结构上、字符种类上、语义信息上的多种修改方式。为了捕捉复杂的口令重用行为，学术界已经提出了多种口令重用模型，但均未有效运用于蜜口令库方案中。究其根本，对于蜜口令库方案而言，口令重用模型必须满足如下条件，才能够在蜜口令库方案场景下有效捕捉重用机制，同时不引入额外的安全风险：

1. **完备性**。对于任意给定的一对用户口令 pw_A 和 pw_B，口令重用机制能够对用户基于口令 pw_A 和 pw_B 的重用行为进行描述并有效建模。例如，对于 abc123 和 123456abc，口令重用机制应当能够有效描述用户基于 abc123，来创建 123456abc 的重用行为。
2. **无歧义性**。对于任意给定的一对用户口令 pw_A 和 pw_B，口令重用机制对用户基于口令 pw_A 和 pw_B 的重用行为有且仅有一种描述方式。例如，对于 abc123 和 123abc，给定口令 abc123，重用机制只有一种方式能够描述修改 abc123 来创建 123456abc 的重用行为。
3. **双向可逆性**。真实口令库按照重用机制进行编码的口令对 $< pw_A, pw_B >$，和随机解码得到的 $< pw_A, pw_B >$，二者不可区分。

无法满足完备性的口令重用机制，将无法有效对用户复杂而多样的口令重用行为进行完整的建模，例如 Golla 蜜口令库方案的重用机制只能捕捉用户对口令尾部进行修改的重用行为。无法满足无歧义性和双向可逆性的口令重用机制，其生成的相当一部分诱饵口令库可以被攻击者根据重用机制的特点进行直接排除。接下来，本文将对学术界现有的口令重用模型进行逐一分析，并对最符合上述条件的口令重用模型进行适当改造，进而设计能够有效捕捉重用行为、不引入额外安全隐患的口令重用机制。

1. **Targuess-II 和 Das 口令重用模型**。

   Targuess-II [22] 和 Das [53] 是学术界最早提出的口令重用模型。其根据一系列启发式方法和统计学模型，对用户的重用行为进行统计分析，进而捕捉用户的重用行为。虽然 Targuess-II 在口令重用攻击中的表现较好，但其启发式重用规则不满足完备性，进而不适合在蜜加密机制中加以使用。例如，Targuess-II 无法捕捉基于 AbCD1234 来创建 AbcD1234 的重用行为。

2. **Pass2Path 和 Pass2Edit 口令重用模型**。

Pass2Path [47] 和 Pass2Edit [51] 使用编辑操作（字符层面的插入、删除、替换）对口令重用行为进行描述，并且使用深度学习方法来学习用户的口令重用行为。然而，编辑操作本身具有歧义的，使得这两个口令重用模型不满足无歧义性，进而不适合作为蜜加密场景下的口令重用机制。例如，对于用户基于 abc123 得到 123123 的口令重用行为，Pass2Path 有两种方式对其进行描述：将 abc 逐字符替换为 123，或者先删除 abc，然后插入 123。这种歧义导致编码时所需要选择的重用方式无法被有效遍历，并且在解码时有可能得到和编码规则不符的重用方式，进而被攻击者直接判定为诱饵口令库。

3. **PassBERT 口令重用模型**。

PassBERT [14] 将口令重用行为视为对原始口令中各个字符进行标记的一系列操作，每个标记对应插入、替换、保留等字符级别的编辑操作。如前文所述，PassBERT 所使用的编辑操作同样不满足无歧义性。此外，PassBERT 不满足完备性，其重用机制只能描述不超过三次插入操作的重用行为，进而无法有效捕捉用户复杂的口令重用行为。

4. **PointerGuess 口令重用模型**。

PointerGuess [48] 并不使用编辑操作或启发式重用操作对口令重用行为进行建模，而是将口令重用行为视为根据已有序列，从头开始重建新序列的过程。在这个过程中，PointerGuess 使用指针网络来捕捉用户在口令修改过程中对旧口令的“保留行为”，并在逐字符重建用户新口令的时候选择何时保留原有口令中的已有部分、何时直接生成新的字符。PointerGuess 的逐字符重建机制是无歧义的，因为任意一种序列从前到后的重建方式有且仅有一种；同时，PointerGuess 具有完备性，因为其序列重建的过程可以根据给定口令，在完整的口令概率空间中进行基于条件概率的口令生成，进而可以有效描述任意两个给定口令的重用过程。

综上所述，PointerGuess [48] 是目前为止学术界唯一能够同时满足完备性和无歧义性的口令重用模型，本文的蜜口令库方案将以 PointerGuess 为基础，构建蜜口令库方案中的口令重用机制和编码解码方法。

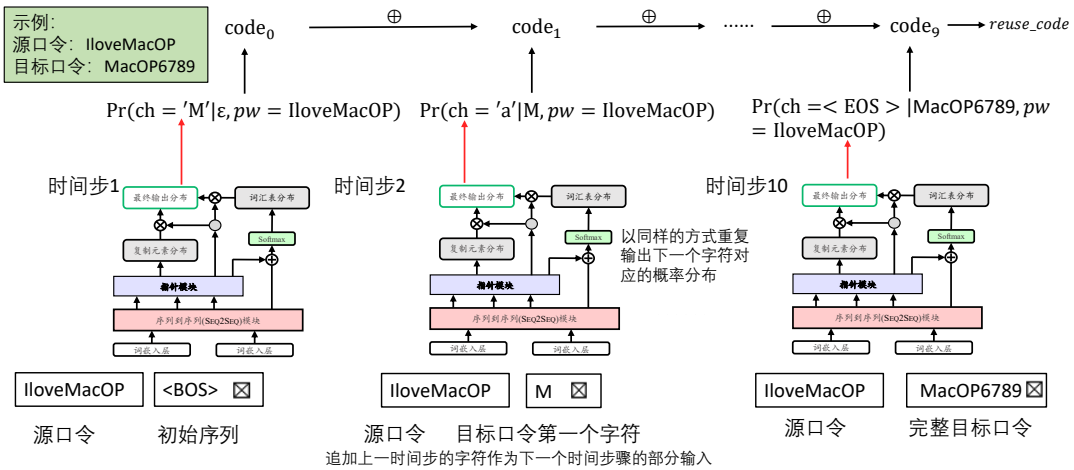


图 4.9 PointerGuess 的模型架构

4.6.2 逐字符的重用口令编码方法

PointerGuess[48] 模型中，用户的口令重用行为可以被描述为“拷贝”原始口令中部分字符，同时“选用”新字符，最终从头到尾构造完整字符序列的过程。PointerGuess 在训练过程中，得到用户根据自己的旧口令，“拷贝”字符或“选用”新字符的概率，进而学习用户根据旧口令选用新口令的重用行为。具体来说，为了灵活决定是复制旧口令中的字符还是直接生成新字符，PointerGuess 使用的软开关机制 p_g：

$$p_g = \sigma\left(W_c \cdot c_t + W_s \cdot s_t + W_y \cdot y_t + b_g\right), \tag{4.22}$$

其中，$\sigma(\cdot)$ 为 sigmoid 函数，W_c、W_s、W_y、b_g 对应模型中的线性层，由训练过程中线性层所学习的参数决定；y_t 是解码器在第 t 个时间步的输入。最后，PointerGuess [48] 将 P_{vocab} 和 P_{copy} 整合生成 P_{gen}，表示口令的最终条件概率 P_{gen}:

$$Pr_{\text{gen}}(c) = p_g \cdot Pr_{\text{copy}}(c) + (1 - p_g) \cdot Pr_{\text{vocab}}(c). \tag{4.23}$$

这使得 PointerGuess 能够在预测口令的下一字符时，按照其学习到的重用行为，从旧口令中复制字符，或者直接从字符集里选择新字符。这种逐字符的生成方式，可以很好地和密口令库方案中的蜜加密技术结合起来。如图4.9所示，对于重用口令，首先将其源口令输入到 PointerGuess 模型中作为条件化，然后逐字符地从该模型所输出的条件概率空间中，按照目标口令每个字符所对应的条件概率，通过 DTE 进行编码，得到编码空间中每个口令所对应的编码。在对口令库里的重用口令进行解码时，过程则与上述过程相反，按照蜜加密技术的解

码步骤逐步进行即可，此处不再赘述。

4.6.3 基于条件概率的口令重用判定方法

本研究采用基于 PointerGuess 条件概率分布的口令重用判定机制，对用户口令库中哪些口令为重用口令进行判定，并且对这些口令进行重用编码，记录重用口令的下标。这样一来，真实口令库和诱饵口令库中所呈现的重用口令位置、重用口令特征将会更加相似，同时避免了启发式重用判定机制所面临的重用行为捕捉能力不足问题。

此处首先考虑对任意两个给定的不相同口令，如何判定其中一个是否为另一个的重用口令。理论上，一对真实的用户重用口令和一对从口令分布中随机采样的独立口令，它们在 PointerGuess 重用模型下计算得到的条件概率应当遵循两个不同的分布，因此可以通过贝叶斯后验概率来量化一对口令属于重用口令对（而非独立口令对）的可能性。形式上，设 $\mathcal{M}_\theta$ 为预训练的 PointerGuess 重用模型。给定任意口令对 (pw_A, pw_B)，模型计算条件对数似然得分：

$$s(pw_A, pw_B) = \log P_\theta(pw_B \mid pw_A) = \sum_{k=1}^{|pw_B|} \log P_\theta(c_k^{(B)} \mid c_{<k}^{(B)}, pw_A). \tag{4.24}$$

此时，假定重用口令对的条件概率与独立口令对的条件概率分别服从两个不同的分布。真实重用口令对 $\langle pw_{\text{src}}, pw_{\text{reu}} \rangle$ 的条件概率所服从的分布为 $f_{\text{R}}(x)$；而从口令分布中随机独立采样的口令对 $\langle pw_A, pw_B \rangle$ 的条件概率所服从的分布为 $f_{\neg\text{R}}(x)$。那么当 f_{R} 与 $f_{\neg\text{R}}$ 已知时，对任意口令对，给定其条件概率得分 s，由贝叶斯公式可计算该口令对属于重用类的后验概率：

$$P(\text{R} \mid s) = \frac{\pi_{\text{R}} f_{\text{R}}(s)}{\pi_{\text{R}} f_{\text{R}}(s) + \pi_{\neg\text{R}} f_{\neg\text{R}}(s)}, \tag{4.25}$$

其中 π_{R} 与 $\pi_{\neg\text{R}}$ 分别为重用类与非重用类的先验概率。

在实际计算上述概率时，f_{R} 与 $f_{\neg\text{R}}$ 需要从训练数据中进行估计。为此，本研究分别构造两类数据集：$\mathcal{D}_{\text{R}}$ 由真实重用口令对的条件概率组成，$\mathcal{D}_{\neg\text{R}}$ 则由从口令分布中随机采样的独立口令对的条件概率组成。然后，即可采用高斯核 $K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$ 在对数空间上进行核密度估计（KDE），带宽 h 由 Silverman 法则确定：

$$\hat{f}_{\mathrm{R}}(x)=\frac{1}{|\mathcal{D}_{\mathrm{R}}|h}\sum_{s\in\mathcal{D}_{\mathrm{R}}}K\left(\frac{x-\log_{10}s}{h}\right), \tag{4.26}$$

$$\hat{f}_{\neg\mathrm{R}}(x)=\frac{1}{|\mathcal{D}_{\neg\mathrm{R}}|h}\sum_{s\in\mathcal{D}_{\neg\mathrm{R}}}K\left(\frac{x-\log_{10}s}{h}\right). \tag{4.27}$$

将式4.26和4.27代入式4.25，即得可计算得到任意口令对为重用口令的贝叶斯后验概率。此时，在先验概率的设定上，本文选取 $\pi_{\mathrm{R}}=0.8$，$\pi_{\neg\mathrm{R}}=0.2$[23, 49, 58]。考虑到 PointerGuess 随机解码过程中存在两个口令完全相同的可能，即便对于完全相同的口令对，也以一定概率将其标记为非完全重用。这一概率由 PointerGuess 随机解码中两个口令恰好相同的概率 $\varepsilon=0.018$ 确定，即完全相同口令对中有 1.8% 的概率不被判定为完全重用。

在口令库层面，给定用户口令库 $V=(pw_1,\ldots,pw_n)$，需要对其中每个口令逐一判定其是否为前驱口令的重用，并记录重用索引，以使得真实口令库和诱饵口令库中重用口令的编码模式保持一致。形式上，对每个目标口令 pw_i（$i\geq 2$），首先通过 PointerGuess 模型计算其与所有前驱口令的得分向量 $\mathcal{S}_i=\{s_{j,i}=s(pw_j,pw_i)\mid 0\leq j<i\}$，并通过式4.25得到后验概率向量 $\mathbf{p}_i=(p_{j,i}=P(\mathrm{R}\mid s_{j,i}))_{j=0}^{i-1}$。

记 $\mathcal{E}_i=\{j<i\mid pw_j=pw_i\}$ 为与当前口令完全相同的前驱口令下标集合。以 $\mathbf{p}_i$ 为权重进行加权随机采样，得到候选重用索引 j^* 及对应的后验概率 $p^*=p_{j^*,i}$。设 $u,v\sim\mathrm{Uniform}(0,1)$ 为独立随机数，最终的重用索引 r_i 由以下规则决定：

$$r_i=\begin{cases}\mathrm{RandomChoice}(\mathcal{E}_i), & \mathcal{E}_i\neq\varnothing\wedge v>1-\varepsilon,\\ j^*, & (\mathcal{E}_i=\varnothing\vee v\leq 1-\varepsilon)\wedge u<p^*,\\ -1, & \text{否则},\end{cases} \tag{4.28}$$

其中 $r_i=-1$ 表示该口令独立生成，不重用任何口令库中的前驱口令。粗略地来说，上述口令重用判定机制可以描述为：若存在与当前口令完全相同的前驱口令，则以 $1-\varepsilon=98.2\%$ 的概率直接复制其索引作为重用索引；否则，以贝叶斯后验概率 p^* 作为重用概率，进行随机化的重用判定。其余情况均标记为独立口令，不存在重用情况，即该口令由口令概率模型进行编码和解码，不使用 PointerGuess 模型进行重用编码、解码。

第七节 实验设计与结果

4.7.1 贴合实际攻防场景的实验设计

表 4.2 蜜口令库方案攻防场景设计

场景	语言	防御者 训练数据	服务 类型	口令库 数目	攻击者 训练数据	服务 类型	口令库 数目	用户 口令库	服务 类型	口令库 数目
#1	中文	7k7k CSDN Mop Ys168 Youku	游戏 技术论坛 游戏 分享平台 视频	2,253,247	17173 Netease Sohu Sina Taobao	游戏 邮箱 论坛 社交媒体 电子商务	34,597,431	Ispeak 126 Tianya Renren Dodonew	游戏 邮箱 论坛 社交媒体 电子商务	12,423,692
#2	英语	Datpiff Couponmom Clixsense Ixigo Twitter	音乐 优惠服务 调研服务 旅游 社交媒体	391,1907	Yahoo Dailyquiz Mate1 Duelingnetwork Imgur	门户网站 教育 网上约会 游戏 社交媒体	179,6998	Yahoo voice MathWay Brazzers Warmane Lookbook	语音通信 教育 成人网站 游戏 社交媒体	127,674
#3		与场景 #2 相同						Pastebin	N/A	276

* Pastebin 数据集是迄今为止唯一公开可获取的真实用户口令库数据集。

本文所设计的蜜口令库方案攻防场景如表4.2所示。为了让实验中的场景与现实世界中的攻防场景尽可能贴近，本研究在设计上述场景时的基本考量和原则如下：

- **防御者、攻击者训练数据中的口令分布和用户口令库分布存在差异**。在实际攻防场景中，防御者和攻击者几乎不可能事先得到用户口令库中的口令分布：防御者在训练口令概率模型之前，不可能获得用户明文口令库；攻击者在实施在线验证和区分攻击之前，很难确定用户的明文口令库。因此，在设计上述场景时，本研究使用了完全不同的数据集，分别聚合得到了防御者、攻击者用于训练的口令库和用户口令库。
- **攻击者的训练数据和防御者存在区别**。Cheng 等人假设攻击者能够获取防御者用于训练口令概率模型的完整数据集，并将其作为攻击者的优势进行安全性分析[102]，认为攻击者用于实施区分攻击的辅助数据集应当与防御者的训练数据集保持完全一致。这一设定实际上低估了攻击者能力，因为攻击者能够获取完整 NLE 这一合理假设，就已经蕴含了“攻击者能够获得防御者训练集分布”的攻击者优势。因此，在本章的实验设定中，攻击者不仅能获得完整的白盒 NLE，而且还可以使用和防御者存在差别的额外口令数据集。这样一来，攻击者所使用的口令数据集和能获取的口令分布比防御者更加广阔，能够更好刻画攻击者在攻防场景中的后手优势。
- **攻击者的训练数据比防御者更加贴近用户口令库**。虽然攻击者无法直接获得用户口令库的分布，但是由于用户口令库中的网址、网站相关信息并没有使用蜜加密技术加以保护，攻击者可以针对性地在海量泄露口令数据集

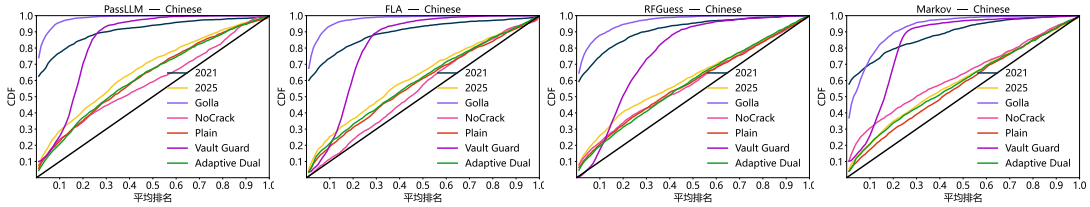
中，获取与用户口令账户所属网站、服务类型等特点更加接近的口令数据集作为辅助数据集。

4.7.2 分布区分攻击下的安全性分析

本文利用 PassLLM、RFGuess、FLA 和 Markov 四种口令概率模型来对蜜口令库方案进行分布区分攻击。形式上，在分布区分攻击中，攻击者通过所获取的口令数据集来逼近用户真实口令库中的口令分布，并直接利用蜜口令库中的口令概率模型来精准获取诱饵口令库的遵循的口令分布。形式上，该攻击所使用的评分函数为：

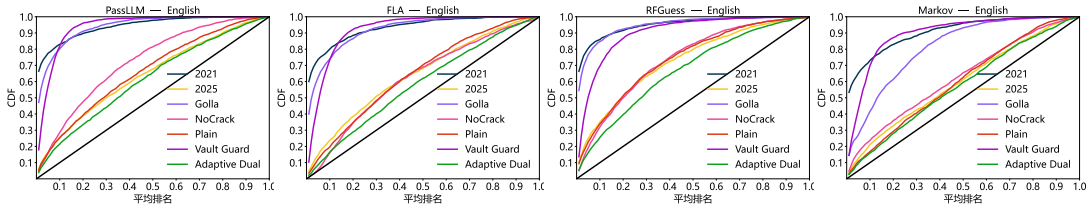
$$s_{DistAware}(v) = \prod_{pw \in v} \frac{\Pr_{PM}(pw)}{\Pr_{NLE}(pw)} \tag{4.29}$$

其中，$\Pr_{PM}(pw)$ 表示攻击者口令概率模型 PM 对口令 pw 分配的概率。该评分函数实质上使用口令概率模型 PM 来对真实口令库中的口令分布进行近似，利用真实、诱饵口令库中各个口令在分布上的差别实施区分攻击。因此，在实验中，口令概率模型均使用4.2中的攻击者数据集进行训练。



(a) PassLLM 攻击　(b) FLA 攻击　(c) RFGuess 攻击　(d) Markov 攻击

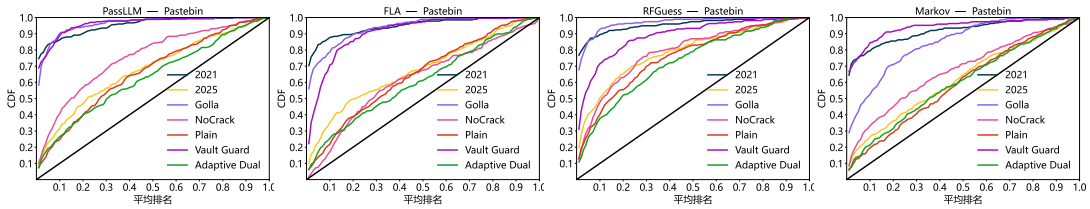
图 4.10 中文聚合口令库上对各蜜口令库方案进行的分布区分攻击结果



(a) PassLLM 攻击　(b) FLA 攻击　(c) RFGuess 攻击　(d) Markov 攻击

图 4.11 英文聚合口令库上对各蜜口令库方案进行的分布区分攻击结果

各蜜口令库方案在三种攻防场景中抵御分布区分攻击的实验结果如图4.10、4.11、4.12所示，方案对应的曲线越接近黑色直线则方案越安全。同时，各方案在分布区分攻击下的平均排名以及安全性下界对应的平均排名如表4.3所示，平



(a) PassLLM 攻击 (b) FLA 攻击 (c) RFGuess 攻击 (d) Markov 攻击

图 4.12 Pastebin 数据集上对各蜜口令库方案进行的分布区分攻击结果

表 4.3 分布区分攻击下的各方案平均排名值

攻击模型	VQ 自适应			无 VQ 自适应			NoCrack			Golla			Cheng'21			Cheng'25			VaultGuard		
	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa
PassLLM	0.377	0.383	0.369	0.378	0.339	0.338	0.413	0.276	0.246	0.017	0.060	0.040	0.090	0.058	0.047	0.343	0.358	0.322	0.162	0.061	0.038
FLA	0.409	0.417	0.392	0.423	0.356	0.355	0.459	0.379	0.385	0.025	0.080	0.065	0.104	0.068	0.053	0.386	0.356	0.325	0.187	0.080	0.094
RFGuess	0.418	0.329	0.273	0.419	0.262	0.242	0.419	0.264	0.215	0.038	0.054	0.033	0.088	0.049	0.044	0.389	0.272	0.212	0.237	0.098	0.112
Markov	0.418	0.425	0.401	0.439	0.401	0.409	0.378	0.376	0.333	0.076	0.188	0.156	0.115	0.093	0.084	0.408	0.402	0.382	0.170	0.104	0.061
最佳攻击	0.377	0.329	0.273	0.378	0.262	0.242	0.378	0.264	0.215	0.017	0.054	0.033	0.088	0.049	0.044	0.343	0.272	0.212	0.162	0.061	0.038
平均偏差	0.174			0.206			0.214			0.465			0.440			0.225			0.413		

* Pa 表示在 Pastebin 数据集上的实验结果，其为迄今为止唯一公开可获取的真实用户口令库数据集。

均排名越接近 0.5 则越安全。在分布区分攻击下，本文所提出的 VQ 深度学习自适应蜜口令库方案在安全性上超越了所有学术界现有蜜口令库方案，安全性的提升幅度为 12.23% ～ 840.81%，展现了本文提出的按照口令分布对子空间进行划分，进而让诱饵口令在分布上更加贴近真实口令的技术路线的有效性。

相比于统计学蜜口令库方案，即 Cheng'21、Golla、NoCrack、VaultGuard 方案，本研究提出的 VQ 深度学习自适应蜜口令库方案在安全性上分别提升了 441.32%、840.81%、12.23% 和 275.93%。这一结果展示了深度学习蜜口令库方案更加强大的泛化能力，其能够有效为真实口令库中的口令赋予相对合理的概率值，进而让诱饵口令和真实口令在 NLE 的口令概率模型中拥有相对一致的概率。相反，统计学模型更加容易在训练集上过拟合，一旦用户口令库中的口令没有在训练数据中出现过，或者用户口令分布和训练集口令分布存在较大差别，分布区分攻击就可以轻易将诱饵口令库和真实口令库区分开来。

相较于不使用自适应机制的深度学习蜜口令库方案，即基于 Transformer 模型的 Cheng'25 方案，本研究提出的 VQ 深度学习自适应蜜口令库方案实现了 18.45% 的安全性提升。该实验结果说明，即便深度学习模型具有泛化能力较强的优势，在攻击者具有不对称信息优势，且用户口令分布和训练数据分布存在差别的情况下，从口令概率分布出发设计自适应机制有助于提升蜜口令库方案的安全性。此外，在消融实验中，本研究去除了 VQ 自适应机制并进行了安全性评估，其抵御分布区分攻击的安全性降低了 10.96%，进一步说明了在口令概

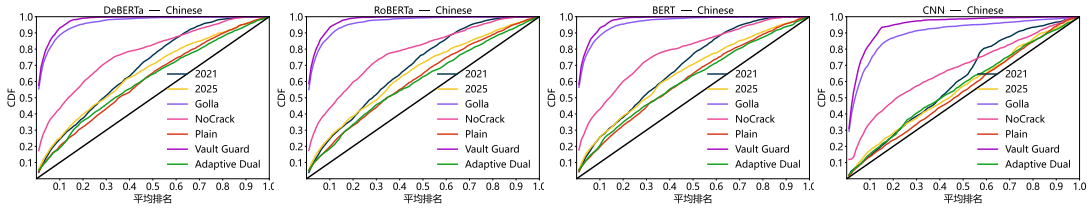
率分布上使用自适应机制的必要性和有效性。

4.7.3 语义区分攻击下的安全性分析

本文利用 DeBERTa、RoBERTa、BERT 和 CNN 四种文本分类模型模型来对蜜口令库方案进行语义区分攻击。在语义区分攻击中，攻击者通过所获取的口令数据集，以及从 NLE 中采样得到的一系列诱饵口令，将二者作为两个类别的口令来训练语义分类器模型。形式上，该攻击所使用的评分函数为：

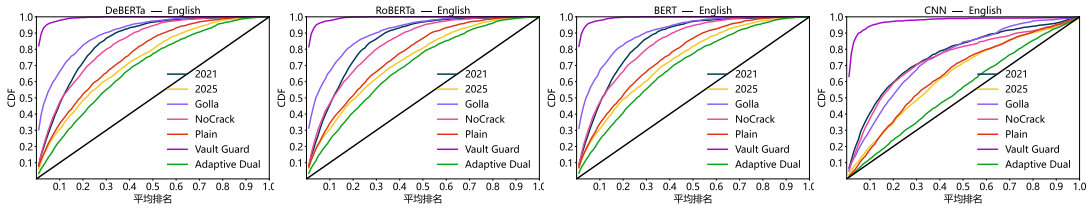
$$s_{SemanticAware}(v)=\prod_{pw\in v}\frac{\Pr_{\mathcal{C}}^{real}(pw)}{\Pr_{\mathcal{C}}^{fake}(pw)}=\prod_{pw\in v}\frac{\Pr_{\mathcal{C}}^{real}(pw)}{1-\Pr_{\mathcal{C}}^{real}(pw)} \tag{4.30}$$

其中，$\Pr_{\mathcal{C}}^{real}(pw)$ 和 $\Pr_{\mathcal{C}}^{real}(pw)$ 分别表示语义分类器模型 $\mathcal{C}$ 对口令 pw 分类为真口令/诱饵口令的概率。由于我们在区分攻击的排序中只关心不同口令库评分的相对大小，所以该评分函数可以直接由 $\prod_{pw\in v}\Pr_{\mathcal{C}}^{real}(pw)$ 替代。在实验中，语义分类模型均使用4.2中的攻击者数据集口令作为真实口令，并对每个方案在不同训练集下的 NLE 分别采样诱饵口令进行训练，每个 NLE 都有专门的语义分类模型对其进行攻击。



(a) DeBERTa 攻击　(b) RoBERTa 攻击　(c) BERT 攻击　(d) CNN 攻击

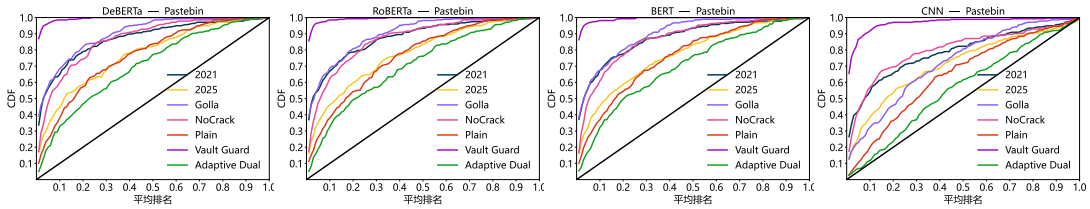
图 4.13 中文聚合口令库上对各蜜口令库方案进行的语义区分攻击结果



(a) DeBERTa 攻击　(b) RoBERTa 攻击　(c) BERT 攻击　(d) CNN 攻击

图 4.14 英文聚合口令库上对各蜜口令库方案进行的语义区分攻击结果

各蜜口令库方案在三种攻防场景中抵御语义区分攻击的实验结果如图4.13、4.14、4.15所示，方案对应的曲线越接近黑色直线则方案越安全。同时，各方案在语义区分攻击下的平均排名以及安全性下界对应的平均排名如表4.4所示。不



(a) DeBERTa 攻击 (b) RoBERTa 攻击 (c) BERT 攻击 (d) CNN 攻击

图 4.15 Pastebin 数据集上对各蜜口令库方案进行的语义区分攻击结果

表 4.4 语义区分攻击下的各方案平均排名值

攻击模型	VQ 自适应			无 VQ 自适应			NoCrack			Golla			Cheng'21			Cheng'25			VaultGuard		
	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa
BERT	0.394	0.309	0.300	0.390	0.240	0.240	0.222	0.169	0.135	0.039	0.101	0.102	0.310	0.151	0.125	0.349	0.269	0.230	0.026	0.010	0.008
DeBERTa	0.391	0.313	0.297	0.390	0.240	0.234	0.226	0.170	0.135	0.039	0.101	0.101	0.313	0.152	0.129	0.347	0.273	0.228	0.026	0.010	0.008
RoBERTa	0.402	0.309	0.296	0.388	0.240	0.238	0.223	0.170	0.135	0.039	0.101	0.102	0.311	0.152	0.125	0.349	0.269	0.224	0.025	0.010	0.009
CNN	0.438	0.454	0.437	0.465	0.357	0.358	0.325	0.266	0.218	0.103	0.258	0.278	0.397	0.241	0.215	0.440	0.365	0.290	0.057	0.029	0.034
最佳攻击	0.391	0.309	0.296	0.388	0.240	0.234	0.222	0.169	0.135	0.039	0.101	0.101	0.310	0.151	0.125	0.347	0.269	0.224	0.025	0.010	0.008
平均偏差	0.168			0.213			0.325			0.420			0.304			0.220			0.486		

* Pa 表示在 Pastebin 数据集上的实验结果，其为迄今为止唯一公开可获取的真实用户口令库数据集。

难看出，本文所提出的 VQ 深度学习自适应蜜口令库方案，在抵御语义区分攻击时，安全性相对学术界现有的一系列蜜口令库方案，有了 18.59% ～ 2236.54% 的提升。上述实验结果说明，本文设计的口令语义特征子空间划分方案是有效的。

具体而言，和基于统计学口令概率模型的蜜口令库方案相比（Cheng'21、Golla、NoCrack、VaultGuard 方案），本文设计的 VQ 深度学习自适应蜜口令库方案分别实现了 69.7%、313.61%、89.50% 和 2236.54% 的安全性提升。深度学习模型能够更好捕捉口令中不同尺度的语义特征，进而让随机采样得到的诱饵口令在语义特征上更难和真实口令区分开来。而统计学口令概率模型则很难完整准确对口令语义特征进行建模。例如，n-gram 模型对长距离的口令字符依赖关系几乎束手无策。这就导致统计学模型所解码得到的诱饵口令和真实口令存在一定偏差，容易被基于口令语义特征的分类器辨别出来。

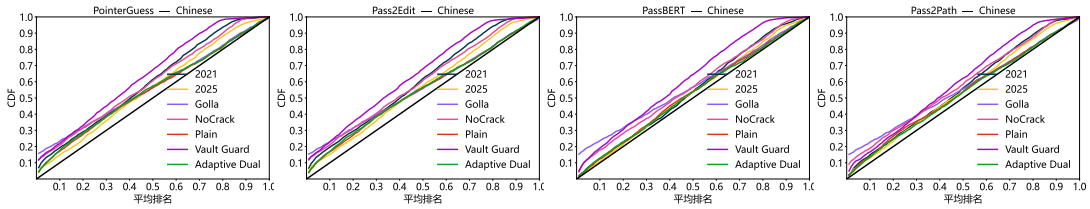
而对于深度学习蜜口令库方案，使用口令语义特征上的自适应机制则能进一步提升其抵御语义区分攻击的安全性。和同样基于深度学习模型的 Cheng'25 方案相比，本研究提出的 VQ 深度学习自适应蜜口令库方案实现了 18.59% 的安全性提升；在消融实验中，去除了 VQ 自适应机制的深度学习蜜口令库方案在安全性上降低了 13.4%。这些结果展现了在口令语义特征上使用自适应机制的优越性。

4.7.4 重用区分攻击下的安全性分析

本文利用 PointerGuess、Pass2Edit、PassBERT 和 Pass2Path 四种口令重用模型来对蜜口令库方案进行重用区分攻击。在重用区分攻击中，该攻击所使用的评分函数为：

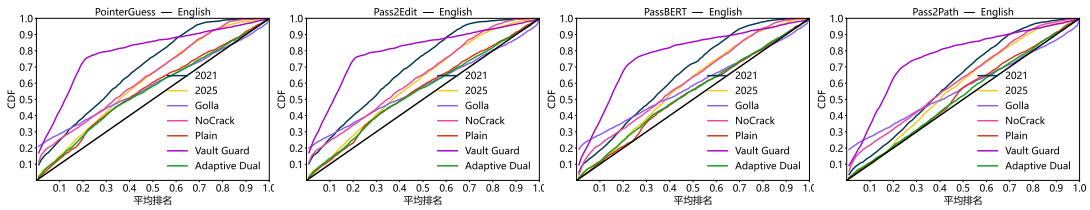
$$
\begin{aligned}
s_{Reuse}(v) &= \prod_{<pw_a,pw_b>\in \mathcal{S}_\nabla} \frac{\Pr_{RealRM}(pw_a|pw_b)}{\Pr_{FakeRM}(pw_a|pw_b)}, \\
\mathcal{S}_\nabla &= \{<pw_a,pw_b> | pw_a,pw_b \in v, sim(pw_a,pw_b) < thresh\},
\end{aligned}
\tag{4.31}
$$

其中，$\mathcal{S}_\nabla$ 表示口令库 v 中，所有满足相似度 $sim(pw_a|pw_b)$ 低于 $thresh$（本文采用编辑距离小于 5 作为判定标准）的口令对 $<pw_a,pw_b>$；$\Pr_{RealRM}(pw_a|pw_b)$ 和 $\Pr_{FakeRM}(pw_a|pw_b)$ 则分别表示在攻击者辅助训练集、诱饵口令库数据集上训练得到的口令重用模型，对口令对 $<pw_a,pw_b>$ 输出的条件概率。



(a) PointerGuess 攻击　(b) Pass2Edit 攻击　(c) PassBERT 攻击　(d) Pass2Path 攻击

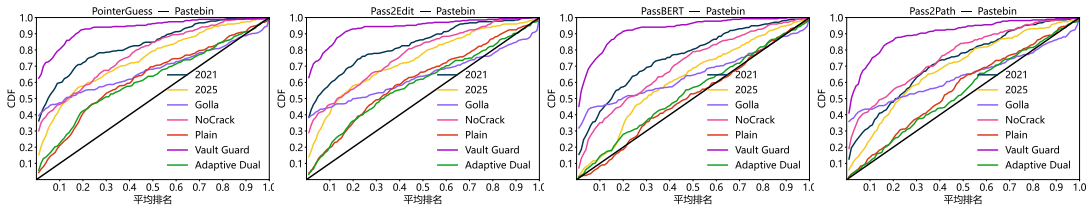
图 4.16 中文聚合口令库上对各蜜口令库方案进行的重用区分攻击结果



(a) PointerGuess 攻击　(b) Pass2Edit 攻击　(c) PassBERT 攻击　(d) Pass2Path 攻击

图 4.17 英文聚合口令库上对各蜜口令库方案进行的重用区分攻击结果

各蜜口令库方案在三种攻防场景中抵御重用区分攻击的实验结果如图4.16、4.17、4.18所示，方案对应的曲线越接近黑色直线则方案越安全。同时，各方案在重用区分攻击下的平均排名以及安全性下界对应的平均排名如表4.5所示。不难看出，本文所提出的 VQ 深度学习自适应蜜口令库方案，在抵御重用区分攻击时，安全性相对学术界现有的一系列蜜口令库方案，有了 9.81% ～ 106.22% 的



(a) PointerGuess 攻击 (b) Pass2Edit 攻击 (c) PassBERT 攻击 (d) Pass2Path 攻击

图 4.18 Pastebin 数据集上对各蜜口令库方案进行的重用区分攻击结果

表 4.5 重用区分攻击下的各方案平均排名值

攻击模型	VQ 自适应			无 VQ 自适应			NoCrack			Golla			Cheng'21			Cheng'25			VaultGuard		
	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa
Pass2Edit	0.449	0.453	0.408	0.451	0.448	0.388	0.398	0.362	0.229	0.435	0.423	0.361	0.396	0.299	0.168	0.436	0.397	0.264	0.349	0.206	0.063
Pass2Path	0.473	0.483	0.465	0.466	0.476	0.441	0.416	0.383	0.241	0.438	0.430	0.359	0.440	0.381	0.277	0.456	0.424	0.315	0.397	0.242	0.101
PassBERT	0.476	0.463	0.461	0.473	0.471	0.491	0.431	0.381	0.296	0.428	0.415	0.351	0.459	0.341	0.241	0.470	0.407	0.382	0.379	0.226	0.074
PointerGuess	0.446	0.445	0.385	0.450	0.438	0.374	0.399	0.355	0.223	0.430	0.412	0.321	0.399	0.305	0.167	0.440	0.384	0.256	0.353	0.208	0.066
最佳攻击	0.446	0.445	0.385	0.450	0.438	0.374	0.398	0.355	0.223	0.428	0.412	0.321	0.396	0.299	0.167	0.436	0.384	0.256	0.349	0.206	0.063
平均偏差	0.075			0.079			0.175			0.113			0.213			0.141			0.294		

* Pa 表示在 Pastebin 数据集上的实验结果，其为迄今为止唯一公开可获取的真实用户口令库数据集。

提升。上述实验结果说明，本文设计的深度学习口令重用机制，在现有各类口令重用模型辅助下的区分攻击中，是相对安全的。

与使用启发式重用机制的蜜口令库方案相比（Cheng'21、Golla、NoCrack、VaultGuard 方案），本研究设计的蜜口令库方案应用了基于深度学习的口令重用编码、解码、判定机制，并且在安全性上分别提高了 47.87%、9.81%、30.66% 和 106.22%。现有方案的启发式重用机制采用了精心设计的启发式重用特征（例如首尾字符修改操作），并且通过真实口令数据集进行了数据驱动的重用设计（例如观察到口令库中有将近 80% 的口令为重用口令）。尽管如此，这些启发式方案对口令重用行为的捕捉能力不足，无法建模更加复杂的口令重用行为，攻击者一旦使用口令重用模型，对真实用户口令重用特征和诱饵口令库中的重用特征进行比较，就可以将诱饵、真实口令库通过重用特征区分出来。而本文所采用的深度学习重用机制不仅能够捕捉任意两个给定口令之间的重用行为特征，不受启发式规则的限制，而且其对口令的重用判定同样是非启发式的。此时，攻击者单从口令库所展现的重用特征上难以将真实口令库区分出来，蜜口令库方案的安全性得到了进一步提升。

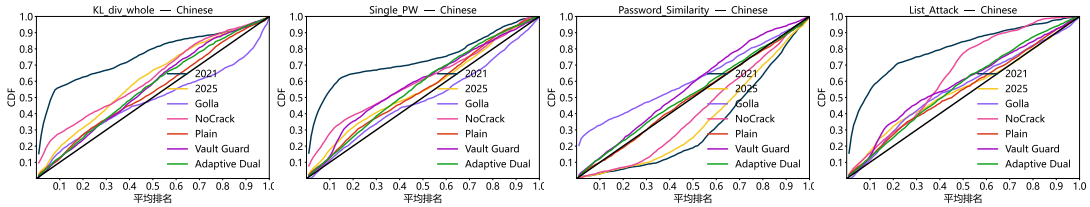
Cheng'25 方案同样使用了基于深度学习的口令重用机制，但该方案的重用机制并非对重用行为的显式建模，而是直接让模型统计整个口令库的语义特征，间接捕捉口令重用行为。这种口令重用建模方案，要求单个深度学习模型在捕捉口令本身的分布之余，进一步建模口令之间的语义相关性，进而捕捉重用行

为。这种重用机制导致口令概率模型需要较大的模型参数量，才能有效捕捉上述特征，体积较大，推理速度较慢，而且重用特征相对单一。相比之下，本研究所设计的深度学习重用机制直接利用口令重用模型对重用行为进行更细粒度的建模，进而实现了 18.52% 的安全性提升。

值得注意的是，本文所提出的重用区分攻击方案，相比学术界现有的重用区分攻击（即口令相似度攻击），实现了更高的攻击效率。这说明在设计区分攻击方案时，有必要利用口令安全研究的前沿技术和攻击方案，将它们迁移到蜜口令库领域，进而实现更加严格、准确的安全性评估。

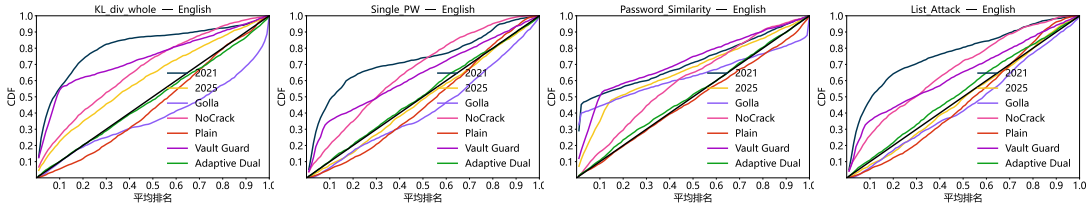
4.7.5 现有区分攻击下的安全性分析

本文利用 KL 散度、单口令、口令相似度和理论最优四种现有的区分攻击方案来对蜜口令库方案进行区分攻击。



(a) KL 散度攻击　(b) 单口令攻击　(c) 口令相似度攻击　(d) 理论最优攻击

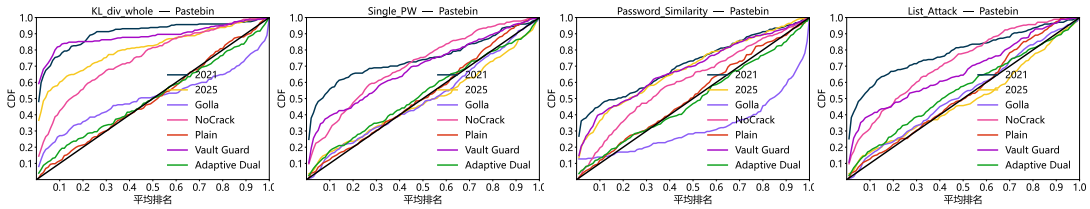
图 4.19 中文聚合口令库上，学术界已有的各类区分攻击下，各蜜口令库方案的安全性。



(a) KL 散度攻击　(b) 单口令攻击　(c) 口令相似度攻击　(d) 理论最优攻击

图 4.20 英文聚合口令库上，学术界已有的各类区分攻击下，各蜜口令库方案的安全性。

各蜜口令库方案在三种攻防场景中抵御现有区分攻击的实验结果如图4.19、4.20、4.21所示，方案对应的曲线越接近黑色直线则方案越安全。同时，各方案在现有区分攻击下的平均排名以及安全性下界对应的平均排名如表4.6所示。可以发现，在学术界现有的一系列区分攻击下，本研究设计的蜜口令库方案同样在安全性上具有很大的优势，实现了 31.48% ～ 177.94% 的安全性提升。现有的



(a) KL 散度攻击 (b) 单口令攻击 (c) 口令相似度攻击 (d) 理论最优攻击

图 4.21 Pastebin 数据集上，学术界已有的各类区分攻击下，各蜜口令库方案的安全性。

表 4.6 学术界现有区分攻击下的各方案平均排名值

攻击模型	VQ 自适应			无 VQ 自适应			NoCrack			Golla			Cheng'21			Cheng'25			VaultGuard		
	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa	中	英	Pa
KL 散度	0.440	0.514	0.495	0.461	0.550	0.496	0.377	0.329	0.251	0.574	0.694	0.615	0.244	0.188	0.096	0.392	0.390	0.192	0.437	0.264	0.113
单口令	0.431	0.493	0.477	0.450	0.539	0.483	0.395	0.342	0.295	0.517	0.582	0.517	0.265	0.272	0.261	0.444	0.520	0.519	0.411	0.366	0.316
口令相似度	0.484	0.490	0.517	0.501	0.522	0.483	0.592	0.396	0.393	0.492	0.408	0.700	0.675	0.286	0.288	0.642	0.337	0.318	0.435	0.279	0.314
理论最优	0.434	0.474	0.451	0.459	0.529	0.471	0.359	0.334	0.283	0.449	0.548	0.479	0.204	0.237	0.222	0.451	0.512	0.511	0.416	0.362	0.331
最佳攻击	0.431	0.474	0.451	0.450	0.550	0.471	0.359	0.329	0.251	0.574	0.694	0.700	0.204	0.188	0.096	0.642	0.337	0.192	0.411	0.264	0.113
平均偏差	0.048			0.043			0.187			0.156			0.337			0.205			0.237		

* Pa 表示在 Pastebin 数据集上的实验结果，其为迄今为止唯一公开可获取的真实用户口令库数据集。

四种区分攻击主要从口令分布或重用机制出发，使用启发式的特征提取方式和计分函数实施区分攻击，上述实验结果说明本文提出的深度学习蜜口令库方案可以很好地抵御已有研究提出的启发式攻击。

此外，和学术界已有的区分攻击相比，本文提出的分布、语义、重用三种区分攻击方案实现了更高的攻击效率。这主要是因为本文提出的三类区分攻击，利用了大量的口令概率模型、口令重用模型和语义分类模型，从口令安全领域的先进成果出发，不再依赖启发式的特征工程方法和计分函数。在本研究提出的三类区分攻击的基础上，随着口令安全研究的不断深入，这些攻击可以用更加先进有效的口令概率模型、重用模型进行实例化，实现更加准确和有效的蜜口令库安全性分析。

第八节 性能分析

蜜口令库方案中，口令概率模型必须在用户本地部署，这对其推理速度和存储空间有一定的要求。接下来，本研究将对各蜜口令库方案所需的存储体积，以及深度学习蜜口令库方案的推理速度进行评估，验证其在实际工作场景中的可部署性。

各蜜口令库方案的总体积如表4.7所示。可以看到，本文提出的 VQ 自适应蜜口令库方案所需的存储开销，低于现有的所有蜜口令库方案。基于统计学口令概率模型的蜜口令库方案需要存储 n-gram 等统计数据，因此模型所需体积普

项目	大小
VQ 自适应蜜口令库方案各组件的总体积	12.1M
NoCrack [34] 方案各组件总体积	100.9M
Golla [35] 方案各组件总体积	100.9M
Cheng'21 [36] 方案各组件体积	493.0M
Cheng'25 [102] 方案各组件体积	187.0M
VaultGuard [49] 方案各组件体积	187.0M

表 4.7 存储开销

遍较大，在实际部署中将给用户带来较大的存储负担。而基于深度学习技术的 Cheng'25 方案则使用 Transformer 模型直接端到端地建模口令库的整体语义信息，因此参数量较大，模型体积也较大。本文的蜜口令库方案分别使用小体积的口令概率模型和口令重用模型，对口令语义和重用特征单独进行建模，降低了存储开销，为进一步在客户端实际部署奠定了基础。

基于统计学模型的蜜口令库方案天然具有较快的推理速度，代价则是需要巨大的存储空间。因此，在推理速度上，本研究不再和基于统计学模型的方案进行对比。在带有单块 Intel Core i9-14900HX 的笔记本电脑上（不使用 GPU 计算资源），本文方案解码得到单个口令所需的平均时间为 171 毫秒，如果一次并行解码多个口令则平均单个口令耗时仅为 5 毫秒。上述结果说明，本文所提出的蜜口令库方案分别使用两个深度学习模型构建口令概率模型和口令重用行为，无需直接使用高参数量的模型在口令库层面进行建模，其所使用的深度学习模型体积较小，推理速度较快，较为适合在算力受限的客户端进行部署。

第九节　本章小结

在本章里，本文提出了基于口令空间划分的自适应机制，并通过量化变分自编码器（VQ-VAE）将口令空间按照分布、语义两类特征切分为了不同的子空间，构建了 VQ 自适应的口令概率模型。该口令概率模型以 VQ 在分布、语义两类特征上对口令映射得到的高维离散向量作为输入，进而条件化地在离散向量所诱导的口令子空间里，按照条件概率分布进行口令的编码和解码，从而让错误主口令解密得到的诱饵口令在分布和语义上和真实口令有一定的相似性。

在自适应口令概率模型之外，本文进一步根据重用口令、独立口令在 PointerGuess 遵循不同条件概率分布的特点，构建了基于 PointerGuess 模型的深度学习口令重用判定机制和诱饵口令库中的口令重用模拟机制。自适应口令概率模型和深度学习重用机制共同构成了本文所提出的完整深度学习自适应蜜口

令库方案，通过深度学习的自适应机制和重用模型，其诱饵口令库能够在语义、分布、重用特征上和真实口令库更加接近，进而更好迷惑攻击者，提升口令库密文的抗猜测性。在大规模数据集上进行的安全性分析实验结果表明，本文提出的蜜口令库方案超越了学术界现有的所有方案，为设计更加安全的蜜口令库方案指明了方向。

第五章 总结与展望

第一节 本文工作总结

1. **使用口令重用模型构造了非定向口令强度评估器**。在口令数据集上进行大量统计和观察之后，本文提出使用脆弱口令的行为可以被视为是一种重用流行口令的行为。因此，使用口令重用模型对这种基于流行口令的重用行为进行捕捉，即可通过重用模型在流行口令诱导的条件概率空间，来对口令在非定向口令猜测攻击下的抗猜测性进行有效评估。在上述技术路线的基础上，本文将 Pass2Edit 重用模型改造为 EditPSM 非定向口令强度评估器，并和现有的一系列先进口令强度评估器在准确性上进行比较。实验结果表明，本文提出的 EditPSM 准确性达到了学术界和工业界口令强度评估器中的先进水平，展现了本文所提出技术路线的有效性。
2. **利用连续学习技术设计了多功能口令强度评估框架**。本文通过对口令重用行为的进一步分析，发现用户对旧口令的重用行为比基于流行口令的重用行为更加复杂。基于这一发现和对连续学习技术的理论分析，本文敲定了将迁移学习技术应用于口令强度评估时的一系列关键设计细节，并提出 VersaPSE 多功能口令强度评估框架。VersaPSE 可以利用现有的口令重用模型，将其先在旧口令重用行为上进行训练，冻结部分参数后继续在基于流行口令的重用行为上进行微调，让口令重用模型能同时建模两种重用行为，进而评估口令在非定向和口令重用两种猜测攻击下的强度。本文利用 VersaPSE 框架将现有的五款口令重用模型进行改造，所得到的五个多功能口令强度评估器均在两种不同攻击场景下分别实现了比现有口令强度评估器更高的准确性。
3. **深度学习自适应蜜口令库方案**。本文从口令的分布和语义两种特征出发，提出了利用这些特征、对口令空间进行划分的自适应蜜口令库方案。通过量化变分自编码器（VQ-VAE）可以将样本在特征空间进行隐式聚类的特性，本文使用 VQ-VAE 构建了条件化的口令概率模型，其能够根据分布、语义两类特征上的高维离散向量，在其诱导的口令子空间中采样和真实口令在分布、语义上相似的诱饵口令。利用该条件化口令概率模型和基于 PointerGuess 深度学习重用模型的口令重用机制，本文提出了深度学习自适应蜜口令库方案，其在大规模真实口令数据集上所展现出

的安全性，超越了现有的一系列蜜口令库方案，体现了深度学习自适应机制的优越性。

第二节 本文工作的局限性

1. 多功能口令强度评估框架的试验评估仅考虑了在线猜测场景，对非定向猜测中的离线猜测场景缺乏考虑。由于离线猜测场景中攻击者能力、强度评估器准确性衡量指标都和在线猜测场景存在较大差别，本文通过VersaPSE 改造口令重用模型所得到的多功能口令强度评估器，在离线猜测攻击中的有效性仍然有待进一步考察。
2. 深度学习自适应蜜口令库方案，特别是口令空间切分机制缺乏形式化的理论基础。本文对口令空间切分方案中公开信息所导致的攻击者优势、自适应机制所能带来的安全性等问题，均没有深入的理论分析，只有初步的讨论和直观上的解释。事实上，自适应机制本身迄今为止缺乏严格的数学定义，是否能引入安全性、怎样才能带来额外安全性，学术界迄今为止也没有严格的定论。怎样在形式上对自适应机制加以定义和约束，并从数学上分析自适应机制在信息公开、安全性提升上的种种性质，是一个有待解决的问题。

第三节 未来工作展望

结合本文的工作，以下问题仍然值得进一步研究和探索：

- **多口令重用行为在口令强度评估中的应用**。目前所有基于口令重用行为的口令强度评估器，均基于单个的用户姊妹口令（旧口令）进行口令强度评估。尽管我们可以采用简单直接的方式，在能获取用户多个姊妹口令时，逐一计算用户口令和这些姊妹口令两两之间的条件概率，但这种方式在本质上存在不足。例如，用户可能基于两个旧口令”nankai2026” 和”ilovesecurity” 来构造新口令”nankai2026security”，而这种多源的口令重用行为是现有口令重用模型难以捕捉的。如何基于这种多口令重用行为来评估用户口令强度，是一个有价值的问题。
- **多口令重用模型的联合重用攻击**。不知攻，焉知防？在利用多口令重用行为评价口令强度之前，学术界现有的公开发表研究里，迄今未有直接基于多口令重用行为的口令重用模型。这很可能是因为，一方面多口令重用行为是否广泛存在、在用户群体中表现出哪些行为特征等问题均未能得到有效解答，基于多口令重用的口令重用模型构建缺乏牢固的现实基础；另一

方面，如何构造训练数据，进而从中提取有效的多口令重用行为学习表征，是一件非常具有挑战性的任务。从用户行为出发，设计基于多口令重用行为的口令模型，有助于我们理解这类攻击的有效性，进而设计相应的防御措施。

- **主口令条件化的蜜口令库方案**。现有的蜜口令库方案，无一考虑了主口令的强度和语义特征。这是因为蜜口令库方案所使用的蜜加密技术中，在其所使用 PBE 方案下，主口令本身的语义和结构，对加密过程、加密结果并不会出现非平凡的影响。然而，可以想象，用户的主口令本身很有可能与用户口令库中所展现的口令分布、口令语义和结构信息产生关联。例如，用户可能使用”yifeizhang123” 作为主口令，加密包含”yifeizhang2026””nankai22yifei” 等口令的口令库明文。然而，主口令与解密得到的明文口令库有一一对应关系，而现有的 PBE 方案和蜜口令库方案，无法在解密过程中让主口令与明文口令库产生任何语义关联。那么攻击者可以利用这一点，以极大的把握确认任何主口令和明文有明显关联的口令库为真实口令库。如何让主口令对解码得到的明文进行条件化，使二者之间存在语义上的关联，是值得关注的一个研究课题。

附 录

第一节 部分非定向口令强度评估器的设定

RNN-PSM。RNN-PSM[39] 的源代码[1]中包含大量的可选超参数设置选项，并且附带了不同的模型配置文件。为保证实验结果的可复现性，本文在此列出实验中所采用的超参数选择：在实验中，本文遵循 Melicher 等人[39] 对客户端模型的设定，即模型中采用 3 层 LSTM、隐藏层维度为 256，并使用 2 层隐藏层维度为 128 的全连接层。该超参数设定在[21] 中已达到和服务端大体积模型相似甚至更优的口令强度评估准确性，所以使用上述设定有助于保证实验结果的公平性，同时节省算力资源。

Microsoft-PSM。目前，微软在 Edge 浏览器中部署了一款口令强度评估器，并在 Edge 用户注册账户时显示口令强度。迄今为止，本研究暂时无法获取其新版口令强度评估器的具体运行机制。所以本文使用的是旧版 Microsoft-PSM 中的设定，其设定与现有学术界研究中[21, 26] 中所采用和描述的版本相同。

异常口令强度评估器输出的处理。部分 PSM 并非对所有输入口令都输出有效结果。例如，zxcvbn 对仅包含空格的口令不输出有效结果。我们将这些结果替换为零，并据此计算 *WSpearman*。这有助于实现更公平的比较，因为无效输出应被视为此类 PSM 的缺陷，而我们赋予的零值大概率会导致其 *WSpearman* 值下降。

第二节 部分重用口令强度评估器的设定

Vec-PPSM [47] 利用相似度度量和词嵌入来评估口令抵御凭据篡改攻击的强度。对于其所采用的 FastText 模型[91]，除使用对应训练集中用户口令训练 FastText 外，本文还使用 Tianya 数据集作为中文场景的额外训练集、Rockyou 数据集作为英文场景的额外训练集。在对待评估口令计算相似度，进而评估口令强度时，本研究使用原作者提供的源代码[2]。

PR-PSM [48] 采用多种启发式方法来辅助口令强度评估。对于 PR-PSM 所使用的 sentencepiece 模型[117]，本研究分别使用 Tianya 和 Rockyou 数据集训练中文和英文场景下的 sentencepiece 模型。PR-PSM 还使用了模拟猜测排名的启发式计算方法，但其原论文中未明确说明其中“重要性分布”的计算方式[48]。因此，

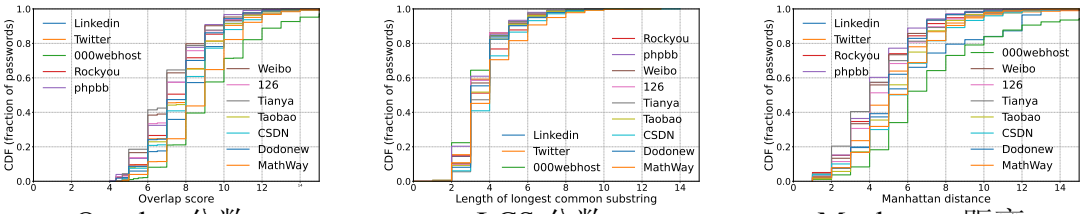
[1] 见https://github.com/cupslab/neural_network_cracking.
[2] 见https://github.com/Bijeeta/credtweak

本研究对 PointerGuess 所生成的 10^3 个口令猜测采用均匀分布的方式进行积分。该设定可能与原工作略有偏差，但本研究发现在实验中，模拟猜测排名机制对最终输出的影响极小（仅影响不到 0.5% 的口令）。因此，本文认为该设定引入的偏差是可以接受的，不会影响性能评估的公平性。

BERT-PSM [14] 在客户端采用条件口令猜测（CPG）和定向口令猜测（TPG）模型。由于 CPG 模型仅对口令中每个字符影响的强度提供可视化提示，它并不影响 BERT-PSM 输出的猜测数。因此，本研究以客户端 TPG 模型输出的猜测数作为 BERT-PSM 的对口令输出的口令强度。在实验中，本研究通过作者提供的源代码，[3]并参照原工作的设定，进一步实现了 PyTorch 版本的 PassBERT、BERT-PSM 模型，以获得更好的性能和兼容性。

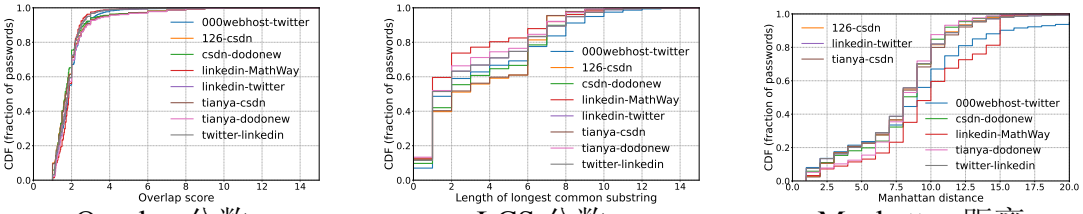
KNN-PSM [58] 利用检索增强生成（RAG）技术来提升 Transformer 模型在口令重用攻击中的准确性，进而提升其口令强度评估的准确性。本研究在实验中使用了作者提供的开源代码[4]。

第三节　用更多相似度指标对用户口令重用行为进行统计



Overlap 分数　　LCS 分数　　Manhattan 距离

图 6.1 各数据集里抽样的用户口令与流行口令的相似度分布。



Overlap 分数　　LCS 分数　　Manhattan 距离

图 6.2 各数据集里用户口令与旧口令的相似度分布。

除了编辑距离和余弦相似度，本文还按照现有研究对口令重用行为进行统计研究的其它常用相似度指标，如 Manhattan 距离、Overlap 分数、LCS（Longest

[3] 见https://github.com/snow0011/PassBertStrengthMeter/tree/main/ClientModel
[4] 见https://github.com/KNNGuess/KNNGuess-code

Common Substrin，最长子串）分数，对口令重用行为进行了统计分析。结果如图6.1和6.2所示，实验结果与先前利用编辑距离、余弦相似度所得到的结果没有本质差别。本文在第三章中所得到的两个统计结论在这三个新指标下依然是成立的。

表 6.1 各口令强度评估器在重用攻击场景下的 AUC 分数

Scenario	VersaPSE (KNNG.)*	VersaPSE (Pass2Edit)*	VersaPSE (PointerG.)*	VersaPSE (PassBERT)*	VersaPSE (Pass2Path)*	KNN-PSM	PR-PSM	BERT-PSM	Vec-PPSM
#1	0.976 (+13.5%∼ +80.8%)	0.967 (+12.4%∼ +79.0%)	0.933 (+8.4%∼ +72.7%)	0.850 (-1.2%∼ +57.3%)	0.960 (+11.6%∼ +77.7%)	0.695 [0.860]	0.718	0.540 [0.703]	0.755 [0.833]
#2	0.950 (+5.7%∼ +56.5%)	0.952 (+5.9%∼ +56.8%)	0.927 (+3.1%∼ +52.7%)	0.886 (-1.4%∼ +46.0%)	0.939 (+4.4%∼ +54.6%)	0.607 [0.894]	0.848	0.636 [0.899]	0.652 [0.784]
#3	0.987 (+4.6%∼ +82.0%)	0.982 (+4.2%∼ +81.2%)	0.973 (+3.2%∼ +79.5%)	0.934 (-0.9%∼ +72.4%)	0.962 (+2.0%∼ +77.5%)	0.542 [0.784]	0.943	0.786 [0.907]	0.738 [0.814]
#4	0.947 (+6.8%∼ +67.6%)	0.943 (+6.3%∼ +66.8%)	0.920 (+3.7%∼ +62.9%)	0.879 (-0.9%∼ +55.5%)	0.931 (+4.9%∼ +64.8%)	0.601 [0.887]	0.799	0.565 [0.866]	0.644 [0.749]
#5	0.986 (+6.3%∼ +71.7%)	0.976 (+5.3%∼ +70.1%)	0.973 (+5.0%∼ +69.5%)	0.947 (+2.2%∼ +65.0%)	0.964 (+4.0%∼ +68.0%)	0.574 [0.802]	0.908	0.815 [0.927]	0.753 [0.845]
#6	0.969 (+6.1%∼ +47.3%)	0.951 (+4.2%∼ +44.6%)	0.940 (+3.0%∼ +42.9%)	0.915 (+0.2%∼ +39.1%)	0.954 (+4.5%∼ +44.9%)	0.803 [0.913]	0.690	0.658 [0.800]	0.761 [0.826]
#7	0.972 (+7.5%∼ +53.8%)	0.962 (+6.4%∼ +52.2%)	0.934 (+3.3%∼ +47.8%)	0.932 (+3.1%∼ +47.5%)	0.959 (+6.1%∼ +51.7%)	0.831 [0.904]	0.673	0.632 [0.759]	0.788 [0.829]
#8	0.950 (+8.0%∼ +34.8%)	0.930 (+5.7%∼ +32.0%)	0.934 (+6.1%∼ +32.5%)	0.890 (+1.1%∼ +26.2%)	0.923 (+4.9%∼ +31.0%)	0.729 [0.880]	0.817	0.750 [0.866]	0.705 [0.827]
Average	0.967 (+11.8%∼ +43.7%)	0.958 (+10.7%∼ +42.3%)	0.942 (+8.9%∼ +39.9%)	0.904 (+4.5%∼ +34.3%)	0.949 (+9.7%∼ +41.0%)	0.673 [0.865]	0.799	0.673 [0.841]	0.724 [0.813]

* 这里为各口令重用模型在 VersaPSE 改造后所产生的口令强度评估器。
圆括号中的百分比为本文改造得到的多功能口令强度评估器，相对现有的口令强度评估器在准确性上的提升幅度。
方括号中的实验结果为现有口令强度评估器在 Zxcvbn-PSM 的协助下，所取得的准确性。

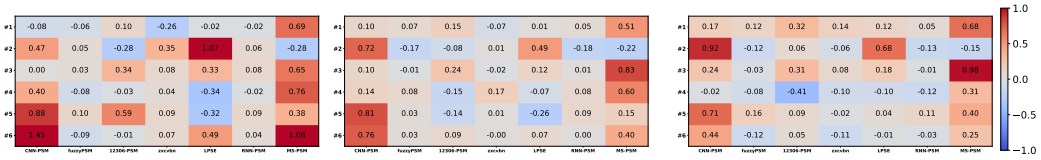
表 6.2 各口令强度评估器在重用攻击场景下的 F1-Score

Scenario	VersaPSE (KNNG.)*	VersaPSE (Pass2Edit)*	VersaPSE (PointerG.)*	VersaPSE (PassBERT)*	VersaPSE (Pass2Path)*	KNN-PSM	PR-PSM	BERT-PSM	Vec-PPSM
#1	0.820 (+8.2%∼ +454.0%)	0.830 (+9.5%∼ +460.9%)	0.726 (-4.3%∼ +390.3%)	0.700 (-7.7%∼ +372.8%)	0.807 (+6.5%∼ +445.6%)	0.559 [0.758]	0.604	0.148 [0.505]	0.388 [0.454]
#2	0.841 (+2.8%∼ +131.7%)	0.858 (+4.9%∼ +136.3%)	0.820 (+0.2%∼ +125.9%)	0.763 (-6.7%∼ +110.2%)	0.840 (+2.7%∼ +131.5%)	0.363 [0.784]	0.818	0.429 [0.805]	0.569 [0.654]
#3	0.953 (+1.4%∼ +502.9%)	0.949 (+1.1%∼ +500.7%)	0.935 (-0.5%∼ +491.5%)	0.873 (-7.0%∼ +452.5%)	0.906 (-3.5%∼ +473.5%)	0.158 [0.731]	0.939	0.727 [0.840]	0.703 [0.740]
#4	0.858 (+12.2%∼ +269.8%)	0.851 (+11.2%∼ +266.7%)	0.826 (+8.0%∼ +256.2%)	0.752 (-1.7%∼ +224.0%)	0.826 (+7.9%∼ +255.9%)	0.340 [0.765]	0.747	0.232 [0.732]	0.567 [0.640]
#5	0.925 (+4.3%∼ +254.4%)	0.919 (+3.6%∼ +251.9%)	0.894 (+0.8%∼ +242.6%)	0.835 (-5.9%∼ +219.9%)	0.868 (-2.2%∼ +232.5%)	0.261 [0.565]	0.887	0.771 [0.859]	0.519 [0.617]
#6	0.864 (+1.7%∼ +79.7%)	0.821 (-3.4%∼ +70.7%)	0.795 (-6.4%∼ +65.4%)	0.743 (-12.6%∼ +54.5%)	0.819 (-3.6%∼ +70.3%)	0.751 [0.850]	0.551	0.481 [0.658]	0.489 [0.580]
#7	0.866 (+1.1%∼ +106.6%)	0.827 (-3.4%∼ +97.3%)	0.769 (-10.2%∼ +83.5%)	0.770 (-10.0%∼ +83.8%)	0.822 (-4.0%∼ +96.2%)	0.790 [0.856]	0.513	0.419 [0.561]	0.518 [0.572]
#8	0.835 (+5.2%∼ +78.0%)	0.823 (+3.6%∼ +75.5%)	0.806 (+1.5%∼ +71.8%)	0.744 (-6.2%∼ +58.7%)	0.800 (+0.7%∼ +70.5%)	0.625 [0.789]	0.757	0.665 [0.794]	0.469 [0.614]
Average	0.870 (+14.2%∼ +80.9%)	0.860 (+12.8%∼ +78.7%)	0.821 (+7.8%∼ +70.8%)	0.772 (+1.4%∼ +60.6%)	0.836 (+9.7%∼ +73.8%)	0.481 [0.762]	0.727	0.484 [0.719]	0.528 [0.609]

* 这里为各口令重用模型在 VersaPSE 改造后所产生的口令强度评估器。
圆括号中的百分比为本文改造得到的多功能口令强度评估器，相对现有的口令强度评估器在准确性上的提升幅度。
方括号中的实验结果为现有口令强度评估器在 Zxcvbn-PSM 的协助下，所取得的准确性。

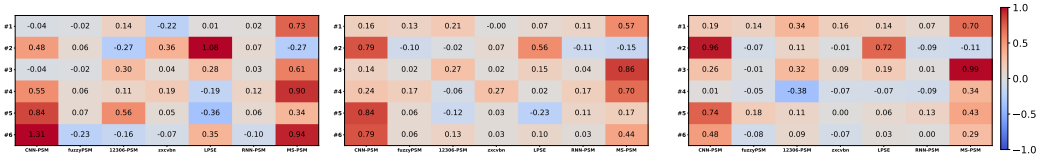
第四节 用其它指标评价重用口令强度评估器的准确性

在第三章中，除了使用平衡准确性来对口令强度评估器在重用攻击下的准确性进行评价，本文进一步使用不平衡任务上常见的 AUC、F1-Score 两类指标来衡量口令强度评估器的准确性。各口令强度评估器在 AUC、F1-Score 两大指



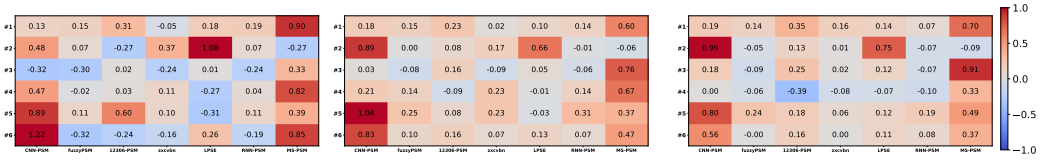
(a) Rank= 10. (b) Rank= 10^2. (c) Rank= 10^3.

图 6.3 VersaPSE(KNNGuess) 在各标志排名的相对 *WSpearman* 分数热力图。



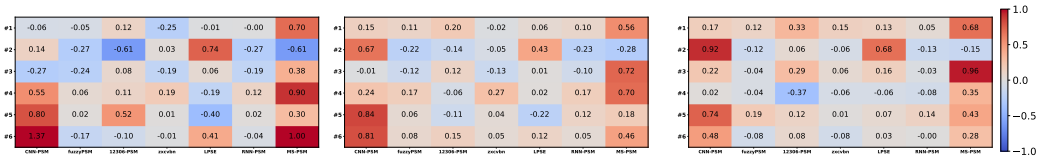
(a) Rank= 10. (b) Rank= 10^2. (c) Rank= 10^3.

图 6.4 VersaPSE(Pass2Edit) 在不同标志排名位置的相对 *WSpearman* 分数热力图。



(a) Rank= 10. (b) Rank= 10^2. (c) Rank= 10^3.

图 6.5 VersaPSE(PassBERT) 在不同标志排名位置的相对 *WSpearman* 分数热力图。



(a) Rank= 10. (b) Rank= 10^2. (c) Rank= 10^3.

图 6.6 VersaPSE(Pass2Path) 在不同标志排名位置的相对 *WSpearman* 分数热力图。

标上的实验结果分别如表6.1和6.2所示。可以看到，使用 AUC、F1-Score 的实验结果，和使用平衡准确性的实验结果，并没有本质区别。事实上，在 AUC 分数上，本文设计的五款多功能口令强度评估器相对现有评估器的准确性提升幅度，比在平衡准确性上还要高得多。这说明本研究提出的 VersaPSE 框架，其改造得到的多功能口令强度评估器在不同准确性指标上，都可以大幅超越现有口令强度评估器，体现了其持续学习技术路线的优越性。

第五节 其它多功能口令强度评估器的 *WSpearman* 热力图

在 PointerGuess 之外，VersaPSE 对 KNNGuess、Pass2Edit、PassBERT 以及 Pass2Path 进行改造，所得到的四款多功能口令强度评估器和现有非定向评估

器之间的相对 *WSpearman* 分别如图6.3、6.4、6.5、6.6所示。可以看到，除了VersaPSE (PointerGuess) 以外，本文的另外四款多功能口令强度评估器同样在非定向口令强度评估场景中，达到了现有口令强度评估器的先进水平。

参考文献

[1] Statista. Number of internet users worldwide from 2005 to 2025, Jan. 2026. https://www.statista.com/statistics/273018/number-of-internet-users-worldwide.

[2] 新华社. 新华社权威快报：我国网民规模达 11.25 亿人, Feb. 2026. https://www.news.cn/politics/20260205/2b0859a2bc4a498ebf0bbaeb07557820/c.html.

[3] FMTAD. Password Statistics 2025: Global Trends & Usage Analysis, Mar. 2025. https://freemindtronic.com/password-statistics-2025-global-trends-usage-analysis/.

[4] Lukas Grigas. Stop reusing passwords: what recent NordPass survey reveals, Apr. 2025. https://nordpass.com/blog/stop-reusing-passwords/.

[5] Dashia Milden. Think Your Password's Safe? Think Again. CNET Survey Reveals 49% of Americans Have Risky Password Habits, July 2025. https://www.cnet.com/tech/services-and-software/password-survey-2025/.

[6] BitWarden. World Password Day Survey 2024, Apr. 2024. https://bitwarden.com/resources/world-password-day-2024/.

[7] Bill Toulas. Have I Been Pwned warns of DatPiff data breach impacting millions, Jan. 2022. https://www.bleepingcomputer.com/news/security/have-i-been-pwned-warns-of-datpiff-data-breach-impacting-millions/.

[8] John Fontana. IEEE admits password leak, Aug. 2024. http://www.zdnet.com/ieee-admits-password-leak-says-problem-fixed-7000004804/.

[9] RockYou2024: 10 billion passwords leaked in the largest compilation of all time, 2024. https://cybernews.com/security/rockyou2024-largest-password-compilation-leak.

[10] Davey Winder. 184,162,718 Passwords And Logins Leaked —Apple, Facebook, Snapchat, May 2025. https://www.forbes.com/sites/daveywinder/2025/05/23/184162718-passwords-and-logins-leaked---apple-facebook-snapchat/.

[11] István F., Micaela A. Which Password Managers Have Been Hacked?, Sept. 2023. https://bestreviews.net/which-password-managers-have-been-hacked/.

[12] TRM Team. TRM Traces Stolen Crypto from 2022 LastPass Breach —On-chain Indicators Suggest Russian Cybercriminal Involvement, Dec. 2025. https://www.trmlabs.com/resources/blog/trm-traces-stolen-crypto-from-2022-lastpass-breach-on-chain-indicators-suggest-russian-cybercriminal-involvement.

[13] Ding Wang, Yunkai Zou, Zijian Zhang, *et al.* Password Guessing Using Random Forest. In: Proc. USENIX SEC 2023: 965–982.

[14] Ming Xu, Jitao Yu, Xinyi Zhang, *et al.* Improving real-world password guessing attacks via bi-directional Transformers. In: Proc. USENIX SEC 2023: 1001–1018.

[15] Tao Yang, Ding Wang. Rankguess: Password guessing using adversarial ranking. In: Proc. IEEE S&P 2025: 682–700.

[16] 郗志红, 周永彬, 李勇, *et al.* 基于深度学习的口令猜测方法的组合优化构造. 信息安全学报, 2026, 10(6): 150–162.

[17] 邹云开, 汪定. 口令猜测研究进展. Journal of Cryptologic Research, 2024, 11(1): 67.

[18] 王平, 汪定, 黄欣沂. 口令安全研究进展. 计算机研究与发展, 2016, 53(10): 2173–2188.

[19] 王传旺, 韩伟力. 面向子词级口令猜测模型的蒙特卡洛强度评估优化算法. 计算机应用与软件, 2025, 42(2): 361–366.

[20] 汪定, 邹云开, 陶义, *et al.* 基于循环神经网络和生成式对抗网络的口令猜测模型研究. 计算机学报, 2021, 44(8): 1519–1534.

[21] Ding Wang, Xuan Shan, Qiying Dong, *et al.* No single silver bullet: Measuring the accuracy of password strength meters. In: Proc. USENIX SEC 2023: 947–964.

[22] Ding Wang, Zijian Zhang, Ping Wang, *et al.* Targeted online password guessing: An underestimated threat. In: Proc. ACM CCS 2016: 1242–1254.

[23] Ding Wang, Debiao He, Haibo Cheng, *et al.* fuzzyPSM: A New Password Strength Meter Using Fuzzy Probabilistic Context-Free Grammars. In: Proc. IEEE/IFIP DSN 2016: 595–606.

[24] Maximilian Golla, Markus Dürmuth. On the Accuracy of Password Strength Meters. In: Proc. ACM CCS 2018: 1567–1582.

[25] Blase Ur, Felicia Alfieri, Maung Aung, *et al.* Design and evaluation of a data-driven password meter. In: Proc. ACM CHI 2017: 3775–3786.

[26] Microsoft Password Checker, Feb. 2012. https://devilsworkshop.org/microsoft-password-checker/.

[27] zxcvbn: realistic password strength estimation, Apr. 2012. https://dropbox.tech/security/zxcvbn-realistic-password-strength-estimation.

[28] 12306 registration page, Aug. 2024. https://kyfw.12306.cn/otn/regist/init.

[29] Bitwarden: How strong is your password? https://bitwarden.com/password-strength/.

[30] Recently Added Breaches, Aug. 2024. https://haveibeenpwned.com/.

[31] Andrew Zangre. COMB data breach: what it means, and how to protect yourself, Feb. 2021. https://blog.1password.com/what-comb-means-for-you-and-your-business/.

[32] Yahoo data breaches, 2016. https://en.wikipedia.org/wiki/Yahoo_data_breaches.

[33] Ari Juels, Thomas Ristenpart. Honey encryption: Security beyond the brute-force bound. In: Proc. EUROCRYPT 2014: 293–310.

[34] Rahul Chatterjee, Joseph Bonneau, Ari Juels, *et al.* Cracking-resistant password vaults using natural language encoders. In: Proc. IEEE S&P 2015: 481–498.

[35] Maximilian Golla, Benedict Beuscher, Markus Dürmuth. On the security of cracking-resistant password vaults. In: Proc. ACM CCS 2016: 1230–1241.

[36] Haibo Cheng, Wenting Li, Ping Wang, *et al.* Incrementally updateable honey password vaults. In: Proc. USENIX SEC 2021: 857–874.

[37] Haibo Cheng, Zhixiong Zheng, Wenting Li, *et al.* Probability model transforming encoders against encoding attacks. In: Proc. USENIX SEC 2019: 1573–1590.

[38] Fei Duan, Ding Wang, Chunfu Jia. A Security Analysis of Honey Vaults. In: Proc. IEEE S&P 2024: 1424–1442.

[39] William Melicher, Blase Ur, Sean M Segreti, *et al.* Fast, lean, and accurate: Modeling password guessability using neural networks. In: Proc. USENIX SEC 2016: 175–191.

[40] Matt Weir, Sudhir Aggarwal, Breno De Medeiros, *et al.* Password cracking using probabilistic context-free grammars. In: Proc. IEEE S&P 2009: 391–405.

[41] Yunkai Zou, Maoxiang An, Ding Wang. Password guessing using large language models. In: Proc. USENIX SEC 2025: 7799–7818.

[42] Edge password health indicator, Aug. 2024. https://support.microsoft.com/en-us/topic/password-health-indicator-5df7b4bc-cdb2-430a-9951-034accc57ff3.

[43] Daniel Lowe Wheeler. zxcvbn: Low-Budget Password Strength Estimation. In: Proc. USENIX SEC 2016: 157–173.

[44] What is zxcvbn.min.js in WordPress? https://webtrainingwheels.com/zxcvbn-wordpress/.

[45] Matteo Dell'Amico, Maurizio Filippone. Monte Carlo strength evaluation: Fast and reliable password checking. In: Proc. ACM CCS 2015: 158–169.

[46] Claude Castelluccia, Markus Dürmuth, Daniele Perito. Adaptive password-strength meters from markov models. In: Proc. NDSS 2012.

[47] Bijeeta Pal, Tal Daniel, Rahul Chatterjee, *et al.* Beyond credential stuffing: Password similarity models using neural networks. In: Proc. IEEE S&P 2019: 417–434.

[48] Kedong Xiu, Ding Wang. PointerGuess: Targeted password guessing model using pointer mechanism. In: Proc. USENIX SEC 2024: 5555–5572.

[49] Zhenduo Hou, Tingwei Fan, Fei Duan, *et al.* How to Design Secure Honey Vault Schemes. In: Proc. ACM CCS 2025: 4319–4333.

[50] Jerry Ma, Weining Yang, Min Luo, *et al.* A study of probabilistic password models. In: Proc. IEEE S&P 2014, 2014: 689–704.

[51] Ding Wang, Yunkai Zou, Yuan-An Xiao, *et al.* Pass2Edit: A Multi-Step Generative Model for Guessing Edited Passwords. In: Proc. USENIX SEC 2023: 983–1000.

[52] Jens Steube. Hashcat, Sept. 2022. https://hashcat.net/hashcat/.

[53] Anupam Das, Joseph Bonneau, Matthew Caesar, *et al.* The tangled web of password reuse. In: Proc. NDSS 2014: 23–26.

[54] Marcus White. Learn what 1 billion+ malware-stolen credentials mean for your 2025 security to-do list, 2025. https://specopssoft.com/blog/report-one-billion-malware-stolen-credentials/.

[55] OWASP: Credential stuffing, 2021. https://owasp.org/www-community/attacks/Credential_stuffing.

[56] 2021 Credential Stuffing Report, 2021. https://www.f5.com/labs/articles/threat-intelligence/2021-credential-stuffing-report.

[57] Nabeel Saeed. Top Insights From Our 2022 State of Secure Identity Report, Sept. 2022. https://auth0.com/blog/top-insights-from-our-2022-state-of-secure-identity-report/.

[58] Zhen Li, Ding Wang. Targeted Password Guessing Using k-Nearest Neighbors.

[59] Yifei Zhang, Zhenduo Hou, Yunkai Zou, *et al.* EditPSM: A New Password Strength Meter Based on Password Reuse via Deep Learning. In: Proc. INSCRYPT 2024.

[60] Alexandra Nisenoff, Maximilian Golla, Miranda Wei, *et al.* A Two-Decade Retrospective Analysis of a University's Vulnerability to Attacks Exploiting Reused Passwords. In: Proc. USENIX SEC 2023: 5127–5144.

[61] Shiva Houshmand, Sudhir Aggarwal. Building better passwords using probabilistic techniques. In: Proc. ACSAC 2012: 109–118.

[62] Dario Pasquini, Giuseppe Ateniese, Massimo Bernaschi. Interpretable probabilistic password strength meters via deep learning. In: Proc. ESORICS 2020.

[63] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, *et al.* Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proc. EMNLP 2014: 25–29.

[64] Qiying Dong, Chunfu Jia, Fei Duan, *et al.* RLS-PSM: A Robust and Accurate Password Strength Meter Based on Reuse, Leet and Separation. IEEE Trans. Inf. Forensics Secur. 2021, 16: 4988–5002.

[65] Yimin Guo, Zhenfeng Zhang. LPSE: Lightweight password-strength estimation for password meters. Comput. Secur. 2018, 73: 507–518.

[66] Twitter Data Breaches: Full Timeline Through 2023, 2023. https://firewalltimes.com/twitter-data-breach-timeline/.

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, *et al.* Attention is all you need. In: Proc. NeurIPS 2017.

[68] Ding Wang, Ping Wang, Debiao He, *et al.* Birthday, name and bifacial-security: understanding passwords of Chinese web users. In: Proc. USENIX SEC 2019: 1537–1555.

[69] Yue Cao, Hao-Ran Wei, Boxing Chen, *et al.* Continual learning for neural machine translation. In: Proc. NAACL 2021: 3964–3974.

[70] Matthias De Lange, Rahaf Aljundi, Marc Masana, *et al.* A continual learning survey: Defying forgetting in classification tasks. IEEE Trans. Pattern Anal. Machine Intell. 2021, 44(7): 3366–3385.

[71] Ziyan Li, Naoki Hiratani. Optimal Task Order for Continual Learning of Multiple Tasks. In: Proc. ICML 2025: 34578–34603.

[72] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, *et al.* The Balanced Accuracy and Its Posterior Distribution. Proc. ICPR 2010: 3121–3124.

[73] John D Kelleher, Brian Mac Namee, Aoife D'arcy. Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies. MIT press, 2020.

[74] Margherita Grandini, Enrico Bagli, Giorgio Visani. Metrics for Multi-Class Classification: an Overview. ArXiv, 2020, abs/2008.05756.

[75] Haibo He, Yunqian Ma. Imbalanced learning: foundations, algorithms, and applications. 2013.

[76] Liyuan Wang, Xingxing Zhang, Hang Su, *et al.* A comprehensive survey of continual learning: Theory, method and application. IEEE Trans. Pattern Anal. Machine Intell. 2024, 46(8): 5362–5383.

[77] Natawut Monaikul, Giuseppe Castellucci, Simone Filice, *et al.* Continual learning for named entity recognition. In: Proc. AAAI 2021: 13570–13577.

[78] Fuli Qiao, Mehrdad Mahdavi. Learn more, but bother less: parameter efficient continual learning. Proc. NeurIPS 2024, 37: 97476–97498.

[79] Hankyul Kang, Gregor Seifer, Donghyun Lee, *et al.* Do your best and get enough rest for continual learning. In: Proc. CVPR 2025: 10077–10086.

[80] Yujun Shi, Li Yuan, Yunpeng Chen, *et al.* Continual learning via bit-level information preserving. In: Proc. CVPR 2021: 16674–16683.

[81] Chao Zhou, Tom Jacobs, Advait Gadhikar, *et al.* Pay Attention to Small Weights. In: Proc. NeurIPS 2025.

[82] Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar. Foundations of machine learning. MIT press, 2018.

[83] Corinna Cortes, Mehryar Mohri, Afshin Rostamizadeh. Generalization bounds for learning kernels. In: Proc. ICML 2010, 2010: 247–254.

[84] Mehryar Mohri, Afshin Rostamizadeh. Rademacher complexity bounds for non-iid processes. In: Proc. NeurIPS 2008.

[85] Thang Doan, Mehdi Abbana Bennani, Bogdan Mazoure, *et al.* A theoretical analysis of catastrophic forgetting through the ntk overlap matrix. In: Proc. AISTATS, 2021: 1072–1080.

[86] Daniel Goldfarb, Itay Evron, Nir Weinberger, *et al.* The Joint Effect of Task Similarity and Overparameterization on Catastrophic Forgetting—An Analytical Model. In: Proc. ICLR 2024.

[87] Wenjin Wang, Yunqing Hu, Qianglong Chen, *et al.* Task difficulty aware parameter allocation & regularization for lifelong learning. In: Proc. CVPR 2023: 7776–7785.

[88] Maorong Wang, Nicolas Michel, Ling Xiao, *et al.* Improving plasticity in online continual learning via collaborative learning. In: Proc. CVPR 2024: 23460–23469.

[89] Myungsik Cho, Jongeui Park, Suyoung Lee, *et al.* Hard tasks first: Multi-task reinforcement learning through task scheduling. In: Proc. ICML 2024.

[90] Michelle Guo, Albert Haque, De-An Huang, *et al.* Dynamic task prioritization for multitask learning. In: Proc. ECCV 2018: 270–287.

[91] Piotr Bojanowski, Edouard Grave, Armand Joulin, *et al.* Enriching word vectors with subword information. Trans. Assoc. Comput. Linguistics, 2017, 5: 135–146.

[92] Sarah Pearman, Jeremy Thomas, Pardis Emami Naeini, *et al.* Let's go in for a closer look: Observing passwords in their natural habitat. In: Proc. ACM CCS 2017: 295–310.

[93] Ameya Hanamsagar, Simon S Woo, Chris Kanich, *et al.* Leveraging semantic transformation to investigate password habits and their causes. In: Proc. ACM CHI 2018: 1–12.

[94] Sean Oesch, Scott Ruoti. That was then, this is now: A security evaluation of password generation, storage, and autofill in browser-based password managers. In: Proc. USENIX SEC 2020: 2165–2182.

[95] Security.org Team. Password Manager Industry Report and Market Outlook (2023-2024), Sept. 2023. https://www.security.org/digital-safety/password-manager-annual-report/.

[96] Matt Kapko. LastPass breach timeline: How a monthslong cyberattack unraveled, Mar. 2023. https://www.cybersecuritydive.com/news/lastpass-cyberattack-timeline/643958/.

[97] Ravie Lakshmanan. Passwordstate Warns of Ongoing Phishing Attacks Following Data Breach, Apr. 2021. https://thehackernews.com/2021/04/passwordstate-warns-of-ongoing-phishing.html.

[98] Brian Krebs. Experts Fear Crooks are Cracking Keys Stolen in LastPass Breach, Sept. 2023. https://krebsonsecurity.com/2023/09/experts-fear-crooks-are-cracking-keys-stolen-in-lastpass-breach/.

[99] Sam Croley. RTX 4090 v6.2.6. Benchmark, Oct. 2022. https://gist.github.com/Chick3nman/32e662a5bb63bc4f51b847bb422222fd.

[100] Briland Hitaj, Paolo Gasti, Giuseppe Ateniese, *et al.* PassGAN: A deep learning approach for password guessing. In: Proc. ACNS 2019: 217–237.

[101] Markus Dürmuth, Thorsten Kranz. On password guessing with GPUs and FPGAs. In: Proc. PASSWORDS 2014: 19–38.

[102] Haibo Cheng, Fugeng Huang, Jiahong Yang, *et al.* Practically secure honey password vaults: new design and new evaluation against online guessing. In: Proc. USENIX SEC 2025: 7781–7798.

[103] Ding Wang, Haibo Cheng, Ping Wang, *et al.* Zipf's Law in Passwords. IEEE Trans. Inf. Forensics Secur. 2017, 12(11): 2776–2791.

[104] Auguste Kerckhoffs. La cryptographie militaire. J. Sci. Militaires, 1883, 9(4): 5–38.

[105] Claude E Shannon. Communication theory of secrecy systems. The Bell system technical journal, 1949, 28(4): 656–715.

[106] Catalin Cimpanu. Hacker leaks 15 million records from Tokopedia, Indonesia's largest online store, May 2020.

[107] GSergiu Gatlan. Hacker leaks full database of 77 million Nitro PDF user records, Jan. 2021. https://www.bleepingcomputer.com/news/security/hacker-leaks-full-database-of-77-million-nitro-pdf-user-records/.

[108] Jacob Devlin, Ming-Wei Chang, Kenton Lee, *et al.* BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. ACL 2019: 4171–4186.

[109] Yoon Kim. Convolutional neural networks for sentence classification. In: Proc. EMNLP 2014: 1746–1751.

[110] Liu Zhuang, Lin Wayne, Shi Ya, *et al.* A Robustly Optimized BERT Pre-training Approach with Post-training. In: Proc. CCL 2021.

[111] Pengcheng He, Xiaodong Liu, Jianfeng Gao, *et al.* DeBERTa: Decoding-Enhanced BERT With Disentangled Attention. In: Proc. ICLR 2021.

[112] Aaron Van Den Oord, Oriol Vinyals, *et al.* Neural discrete representation learning. In: Proc. NeurIPS 2017.

[113] Yongxin Zhu, Bocheng Li, Yifei Xin, *et al.* Addressing representation collapse in vector quantized models with one linear layer. In: Proc. ICCV 2025: 22968–22977.

[114] Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton. Layer Normalization. arXiv preprint arXiv:1607.06450, 2016.

[115] Ethan Perez, Florian Strub, Harm de Vries, *et al.* FiLM: Visual Reasoning with a General Conditioning Layer. In: Proc. AAAI 2018.

[116] William Peebles, Saining Xie. Scalable Diffusion Models with Transformers. In: Proc. ICCV 2023.

[117] T Kudo. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226, 2018.

致 谢

武汉和爱荷华城没有春秋。我人生里第一个秋天、第一个春天，都是在南开大学度过的。春秋气候不定，花开叶落皆是美景。只是春捂秋冻的道理，年年都做得不尽如人意。每度春秋必然中招，在同一个地方反复跌倒。

很少有人的大学生活能够用“波澜壮阔”来形容，我有幸或不幸地成为了其中一份子。回首四年，没有不甘与遗憾，只是当年那个激扬文字的书生，已经成为了消失在时光中的另一个人；环顾四周，看到身边人各自精彩的大学生活，不由得遐想不同的抉择和可能，所能带来的另外几种经历和人生；颔首深思，假如能够重新来过，我仍会坚定地做出同样的选择。

“要做顶天立地的科研。”汪定老师高超的学术水平、浓厚的科研情怀、细致的治学方法，让我深受感召。在局限于具体技术路线设计之时，汪老师教导我，做科研首先要从价值上看，要学会找到自己论文的“生态位”；在计较单个实验结果得失之中，汪老师鼓励我，做实验首先要从场景出发，要学会找到贴近现实情境、对比公平、系统全面的实验设定；在纠结论文遣词造句之际，汪老师点拨我，写论文首先要为读者服务，力求逻辑清晰，提升沟通效率，写作不需要“标新立异”。汪老师是我走上科研之路的领路人，我会带着汪老师一直以来对我的鼓励、支持、指导，继续为祖国和全人类的信息安全事业战斗到底。

“要有秩序建构者的觉悟，作为秩序建构者，天然更累。历史会给予秩序建构者应有的评价。”上万条微信聊天记录，上千个论文修改意见，上百次悟言一室之内，数十封邮件往来，辗转八个学术会议的投稿，见证了我们百折不挠的科学求索；天津、海口、西安，跨越大江南北的征程，书写了我们追求真理的篇章。侯博士扎实的数理基础，让我回避数学、诋毁数学、咒骂数学，最终敬畏数学，甚至恶补数学；侯博士严密的思维逻辑，让我抵触写作、厌恶写作、害怕写作，最终敢于写作，甚至乐于写作；侯博士深厚的学术积累，让我漠视科学、质疑科学、嘲笑科学，最终追随科学，甚至献身科学。感谢侯朕铎博士，听我想方法时天马行空的碎碎念，看我发邮件时毫无逻辑的意识流，改我写论文时张口就来的荒唐言。我以后的日子里，可能再不会有侯博士这样愿意拦着我、惯着我、带着我横冲直撞的好学长、好老师了。

“相信你能够实现自己的理想。”在南开大学计算机学院、密码与网络空间安全学院，我有幸结识了一些求学路上的“同路人”。我们在同一个战壕并肩作战，在每一场风雨中共同成长。在此，感谢南开大学 2025 届计算机学院校友华志远和 2022 级密码与网络空间安全学院房书睿、罗劲，计算机学院张祯颉、鲁恒泽、孔德嵘，以及 2023 级计算机学院刘迪乘、王众，2024 级计算机学院向宇

航、庄雅雯。感谢你们的帮助、鼓励与陪伴，让我在计算机科学与技术专业这个修罗场里步履蹒跚地坚持了四年。祝计算机和密码与网络空间安全学院的大家都能 0 error(s) 0 warning(s)，不论是打代码还是过日子。

“不知我者，谓我士也骄。”知音难觅，所幸在南开大学，有着共同的理想的我们在“知中国，服务中国”的口号下团结一心，彼此也成为了最知心的朋友。感谢南开大学 2024 届数学科学学院校友刘泳、袁敏翔，商学院校友周雪愉，2025 届医学院校友罗华睿和 2022 级金融学院艾祝垚，人工智能学院贺治达，商学院张括，经济学院邱俊轩、桑文祎，以及 2023 级外国语学院郑家辉，历史学院吴镕希、许粲珩，文学院郭钰，2024 级电子信息与光学工程学院张智凌，2025 级马克思主义学院赵祥泽、周恩来政府管理学院唐浩钧。你们一直以来对我的理解、支持、信任，是我四年以来开展工作的最大动力。在此，我向以上同学和三农学社全体成员，致以崇高的敬意和衷心的感谢！

“回忆起过去，又有一种走马灯的感觉。”相识十六年，星辰远征军永远在路上。感谢武汉大学 2022 级遥感信息工程学院索远、东北大学 2022 级信息科学与工程学院邢锦文、材料科学与工程学院程锦岩、软件学院张宇飞，西北工业大学材料学院罗靖恒。将来我们会在航空航天、软件硬件、信息安全各自不同的领域艰苦奋斗、发光发热。十年一瞬，来得太早的幡然醒悟；物是人非，再也回不去的杨家湾和汤逊湖；日复一日，习惯了工位上的运指如飞；年复一年，爱上了写字楼永不熄灭的灯光。不知我们是否仍会在梦中想起，海拔五千米高原上的长河落日，雪山悬崖旁的冰泉冷涩，对着历史书地理书臧否人物、指点江山的豪情壮志，还有站在讲台上意气风发的那个少年。

“科研学术之路上不只是美好、纯粹，还有噩梦、挣扎，以及潜藏在荒谬中的幽默。”特别需要感谢的是华中师范大学的张老师和谢老师，武汉体育学院的谢老师和王老师。你们是我过去、现在、未来每一篇论文，永远无法出现在作者列表中的共同作者，永远无法写在致谢中的经费资助方，永远无法记录在评审意见中的审稿人。感谢你们二十二年如一日地做我的“开心果”，当我的“啄木鸟”，我爱你们。

“恭敬在心，不在虚文。”最后，感谢所有养育、指导、帮助、鼓励、陪伴过我的亲人、老师、同学、朋友，以及名字因故不能出现在上述致谢中的人。

2026 年 4 月，综合实验楼 A315 门外

本科期间的主要工作和成果

已发表的成果：

1. Yifei Zhang, Zhenduo Hou*, Yunkai Zou, Zhen Li, Ding Wang. EditPSM: A New Password Strength Meter Based on Password Reuse via Deep Learning. In Proc. INSCRYPT 2024 (CCF-C, CACR-B), Kunming, China.

已投稿的成果（应双盲审稿要求，隐去论文题目）：

1. Yifei Zhang, Zhenduo Hou*, Ding Wang. [Title]. Submitted to RAID 2026 (CCF-B), Lancaster, United Kingdom.
2. Zhenduo Hou, Ding Wang*, Yifei Zhang. [Title]. Submitted to NDSS 2027 (CCF-A, CACR-B), Seoul, Republic of Korea.
3. Yilin Li, Yifei Zhang, Guozhu Meng*. [Title]. Submitted to NDSS 2027 (CCF-A), Seoul, Republic of Korea.

本科期间获得的主要奖项：

1. INSCRYPT 2024, **最佳论文提名奖**，云南，昆明
2. 中国密码学会，2024 年全国密码技术竞赛，**全国二等奖**，海南，海口
3. 2025 年全国大学生信息安全竞赛作品赛，**全国三等奖**，陕西，西安

本科期间参加的项目：

1. 基于定向口令猜测技术的口令管理器之设计与实现，国家级大学生创新训练项目，一万元，主持

个人简历

基本信息：

姓名： 张逸非
性别： 男
出生日期： 2003 年 12 月 15 日
通信地址： 湖北省武汉市
电话：[在线版此处隐去]
E-mail: yifeizhang@mail.nankai.edu.cn

教育背景：

2010.09-2016.07　华中师范大学附属小学
2014.08-2015.08　Horace Mann Elementary, Iowa City, IA, U.S.　交换生
2016.09-2022.07　华中师范大学第一附属中学
2022.09-2026.07　南开大学　计算机学院　　计算机科学与技术　　学士
2026.07 起　中国科学院信息工程研究所　　网络空间安全　　攻读博士学位