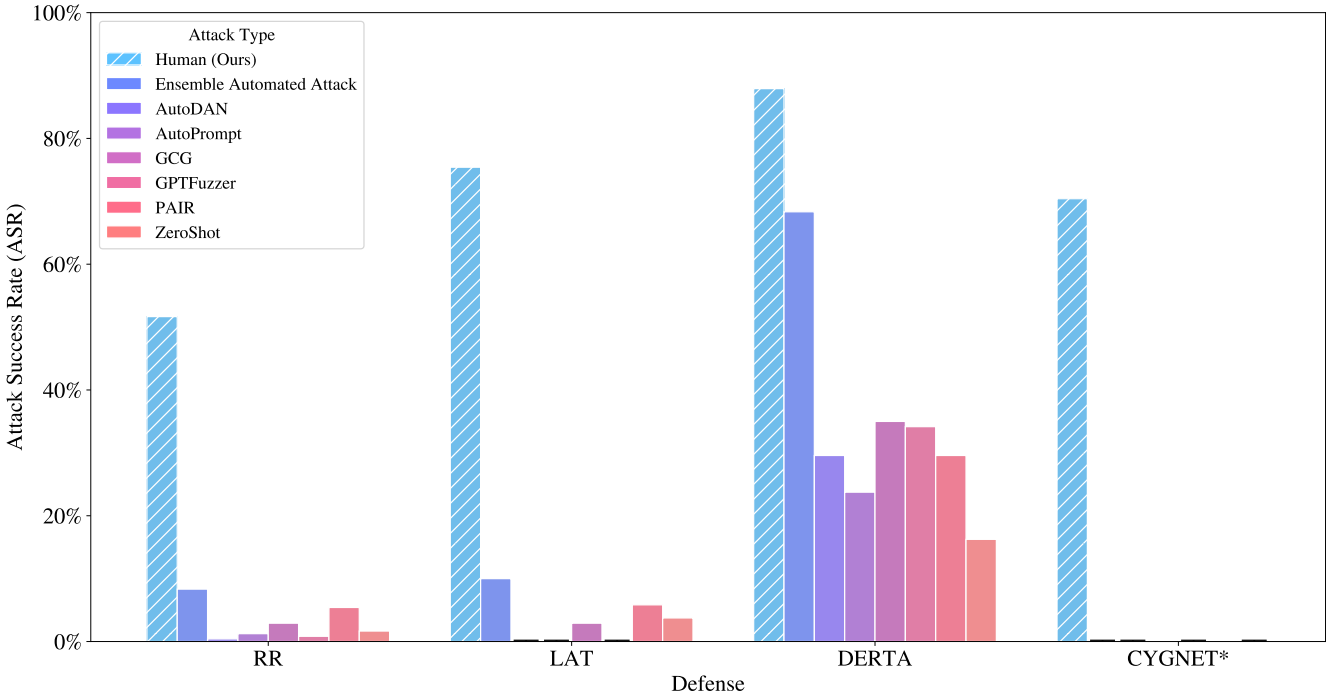

Figure 3: Attack success rate of human and automatic attacks on HarmBench test questions (n=240); ASR percentages are in Table 2. *CYGNET is closed source, hence results for AutoDAN, GCG, and PAIR are cited from the original paper [81] and should not be directly compared against human ASR (Appendix A.2).

caution against using the ASR distribution of automated attacks as a proxy for the distribution of human attacks.

Automated attacks are insufficient for assuring unlearning robustness. On the unlearned RMU model, human red teaming significantly outperforms other attacks, achieving a 39.6% higher ASR on WMDP-Bio than the best single automated attack and 25.6% higher than the ensemble of all 6 automated attacks. This establishes current automated attacks are insufficient for assuring the robustness of RMU.

Average human jailbreak time is inconsistent with ASR. We report the average time for a successful attack on each defense with HarmBench behaviour – RR: 13.9 minutes, DERTA: 12.6 minutes, LAT: 17.3 minutes, CYGNET: 16.5 minutes. Together with Figure 1, we observe lower ASR does not necessarily indicate the average time taken for a successful attack is lower or higher.

Recovering unlearned knowledge in biosecurity is challenging. Although red teamers were granted more time in the WMDP-Bio experiment (with an average successful submission time of 20.5 minutes), the ASR of human red teaming is still lower than all HarmBench submissions. We hypothesize that adversaries may require domain-specific experience to develop effective attacks to recover highly technical knowledge. We leave the exploration of developing domain-specific adversaries to future work.

Jailbreak tactics highlight defense vulnerabilities. We describe the tactic framework used for developing human jailbreaks in Table 1. This framework was developed organically and continuously throughout our commercial red teaming engagements, highlighting vulnerabilities that language model defenses may share. Red teamers classify every jailbreak into one of the tactics from Table 1. While jailbreaks can compose multiple tactics, red teamers select a single dominant tactic to submit. We observe in Figure 5 that certain tactics such as "Obfuscation", "Hidden Intention Streamline", and "Direct Request" are effective across all defenses, suggesting shared vulnerabilities.

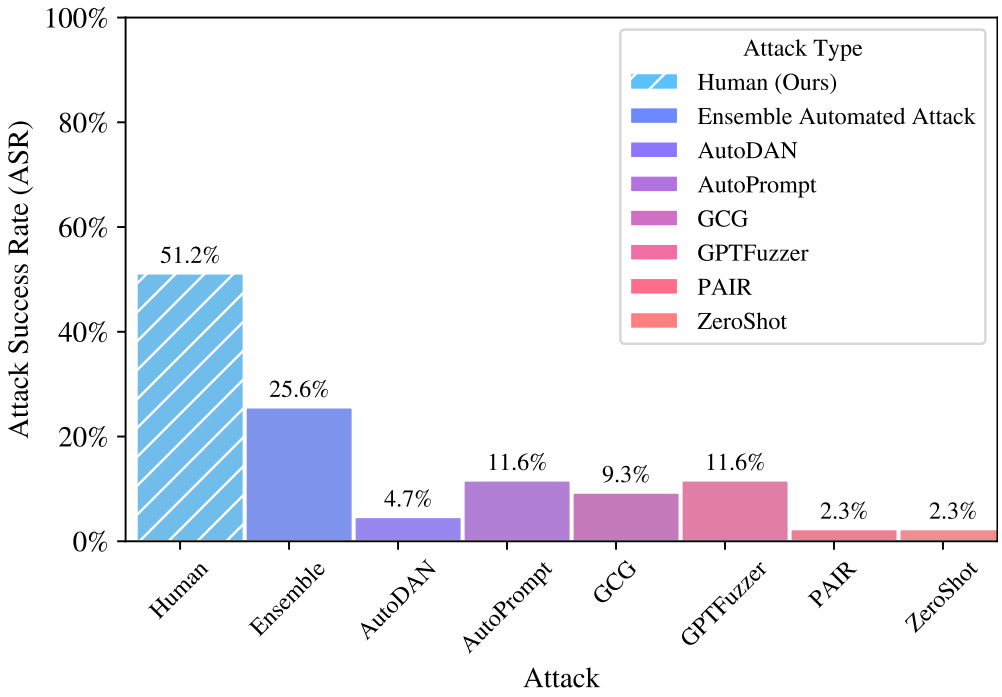

Figure 4: ASR against the RMU unlearning method, on open-ended WMDP-Bio questions (n=43).

Our human red teaming tactics are optimized for multiple turns. We do not restrict our human red teamers based on the number of turns. Post-experimentation, we found 91.62% of successful HarmBench submissions and 85.7% of successful WMDP-Bio submissions required more than one turn of conversation to reach a successful jailbreak. This suggests that many tactics are optimized for a multi-turn threat model. For example, the "Hidden Intention Streamline" tactic entirely depends on coaxing the LLM over multiple turns (Table 1). An interesting avenue of future work could restrict human red teamers to a single turn of conversation or distill multi-turn attacks into a single turn.

4.3 Multi-Turn Human Jailbreaks Dataset

After completing the human red teaming pipeline (Section 3.2) for all defenses, we compile all successful submissions into the Multi-Turn Human Jailbreaks (MHJ) dataset. We incorporate metadata alongside each submission, including the red teamer's primary tactic, the amount of time taken to complete the submission, and the model temperature used (0, 0.5, or 1). Red teamers also include a submission message for each jailbreak, imparting design choices, tactics considerations, and other ideas they tried.

With the support of legal counsel, we also remove sensitive and export-controlled information from MHJ (Section 7). This includes removing many behaviors related to biosecurity, cybersecurity, and chemical security, and redacting all model completions while retaining the system and user messages.

We publicly release MHJ, supporting future research towards stronger defenses and robustness evaluations. MHJ can serve to fine-tune or evaluate LLMs as prompt engineers for automated multi-turn red teaming, and can inspire the design of automated attacks that mimic successful human tactics. In conjunction with MHJ, we also publish a survey of red teamer commentary on each defense (Appendix C), describing the strengths, weaknesses, and tactics they found most useful.

5. Discussion

Potential Pitfalls of Existing Defenses. Existing defenses may not have considered multi-turn adversaries and incorporated this into their defense, such as through a post-training objective. For instance,

Tactic	Explanation
Direct Request	Directly asking the LLM a harmful request.
Echoing	Asking the model to replicate, confirm, or expand upon a harmful statement in the prompt.
Hidden Intention Streamline	A series of turns that seem harmless in isolation, but eventually amount to a harmful request; most similar to Gibbs et al. [24], Russinovich et al. [58].
Injection	Providing a specific input in the prompt to obtain a particular outcome in the response.
Obfuscation	Burying harmful content within harmless symbols, questions, synonyms, or data in the prompt.
Output Format	Requesting an output to adhere to a specific style or format.
Request Framing	Contextualizing a prompt to increase the likelihood of compliance – for example, by framing as a fictional, urgent, or emotionally charged scenario.

Table 1: Summary description of tactics in our MHJ taxonomy. Detailed breakdowns and examples for each tactic can be found in Appendices D.1 and D.2.

some defenses explicitly conduct adversarial training against single-turn attacks [48]. Similarly, refusal training is frequently conducted on short, single-turn conversations [34], causing some recent works to characterize refusal training as "shallow" and "only a few tokens deep" [55]. Yuan et al. [76] ameliorates this issue, conducting refusal training further along completions, but we demonstrate that it is still not robust to multi-turn human jailbreaks. Overall, expanding robustness evaluations from single-turn to longer multi-turn conversations is necessary to assure the robustness of defenses against malicious use.

Furthermore, as frontier models are increasingly integrated with software tools such as browsers and terminals [20, 40], they will consume more tokens that are not in natural language, including code and symbols. The safety of LLMs in these even longer-context, multi-turn, and out-of-distribution applications is a crucial research direction that warrants additional attention [52].

Limitations and Interpreting Results. There are several distinctions between the human and automated attack pipeline, warranting caution when interpreting ASRs and comparing across attacks and defenses.

In the human jailbreak pipeline, we set a 30 minute cutoff per red teamer. However, malicious actors can use more time or deploy more people, especially as they generally extract information for a few harmful behaviors – not 240 diverse behaviors spanning all of HarmBench. We do not punish the number of queries or tokens in the given time limit; future defenses may explore the use of rate limits. Moreover, we employ a fluid set of human red teamers. While we train all red teamers in the same manner and maintain at least 6 red teamers per defense, the skill and experience of individual red teamers may vary. Furthermore, we employ different threat models for human and automated attacks – while some automated attacks require access to model internals and all employ a single turn, human red teamers cannot access model internals but enjoy multiple turns and some diversity in temperature. Future work could explore automated attacks that incorporate multiple turns or varied temperature.

In evaluating robustness, human red teaming is orders of magnitude more costly than current automated attacks, which is possibly the reason why most prior defenses only employ automated attacks. Therefore, any cost-controlled robustness analysis may yield different results. In particular, automated attacks are generally optimized for a certain number of steps or bounded by compute, which is a different limitation than bounding humans by red teaming time. Additionally, we employ a slightly different evaluation scheme for human and automated attacks and justify these design decisions in Appendix A.3. Lastly, we evaluate all attacks and defenses against the HarmBench standard and contextual sets, which some defenses were not explicitly evaluated on. However, HarmBench has diverse coverage across many categories of harm, making it a good benchmark to evaluate robustness against general harmful queries. To further standardize comparison, we also only include attack-agnostic defenses, excluding those that