Fuzzing:Multi-objective optimization

Zhentao Zhu, Zhenyu Wen,

***Abstract*—Height estimation (HE) from high-resolution remote sensing imagery is crucial for 3D scene understanding, with applications such as urban planning and 3D reconstruction. Meanwhile, semantic segmentation (SS) identifies categories and content within these images. Integrating HE and SS can generate more accurate object models, underscoring the need for a unified multi-task framework. Current approaches perform HE and SS independently, overlooking their potential synergy. Therefore, we propose a multi-task network (HEASSNet) to explore the correlation between HE and SS and achieve collaborative learning. Specifically, the two tasks share a pre-trained SAM encoder. We introduce a Low-Rank Adaptation fine-tuning strategy with convolution operations and enhance the local prior in the SAM encoder. The SS branch uses Swin-Transformer to capture global context for qualitative analysis and classification. The HE branch uses convolutional blocks that capture local features to perform quantitative analysis and generate height maps. To learn the distribution of height values, we propose a Feature Enhancement and Aggregation Gate to fuse local and global features, integrating the global context of SS into HE. In addition, the generated height maps provide valuable prompts to the SS branch, distinguishing easily confused categories. Extensive evaluations across multiple datasets validate the effectiveness of our approach.**

***Index Terms*—Semantic Segmentation, Height Estimation, Multi-Tasks Learning, Low-Rank Adaptation.**

I. Introduction

HEIGHT estimation (HE) and semantic segmentation (SS) are fundamental yet challenging tasks in the intelligent interpretation of remote sensing images (RSI). They are critical for various applications, including 3D mapping, urban planning, agricultural management, and forest monitoring. Recent advancements in deep learning have significantly enhanced the performance of both HE and SS. However, the development of remote sensing technologies has introduced the need to tackle complex problems using multi-modal (data or tasks). Despite the individual progress in HE and SS, their potential synergy has often been overlooked. HE provides 3D spatial information, while SS contributes to semantic understanding and component recognition. Each task offers unique perspectives on scene parsing and can implicitly or explicitly inform the other. Consequently, joint learning of HE and SS within a unified network is a promising avenue for research.

Driven by advancements in deep learning, HE and SS have achieved significant progress in remote sensing. For HE, Ghamisi et al. [1] proposed IMG2DSM, a method based on Generative Adversarial Networks (GANs) with skip connections to predict Digital Surface Model (DSM). Paoletti et al. [2] utilized Variational Autoencoders (VAEs) and GANs to generate DSM from single optical images. For SS, significant research has made progress in land cover classification, agricultural monitoring, and urban scene recognition [3]–[5]. Despite these successes, HE and SS have developed independently as separate research. This separation limits the full potential of intelligent interpretation of remote sensing. To address this limitation and enable joint learning of HE and SS, it is necessary to design a multi-task learning (MTL) framework.

The MTL networks are developed to improve performance by leveraging the inherent correlations between tasks. In computer vision, MTL has demonstrated feasibility across various tasks, including SS and depth estimation, SS and object detection, and object detection and classification [6]–[9]. Similarly, in the remote sensing community, MTL has shown promising progress. To extract buildings across various scenes, Guo et al. [10] developed a multi-task framework integrating scene classification and pixel mapping. Feng et al. [11], leveraging computational imaging techniques, proposed a network for super-resolution and colorization. Additionally, substantial advancements have been made in SS and change detection (CD) tasks [12], [13]. Despite the clear relevance and consistency between HE and SS, the potential for joint learning of these tasks remains largely unexplored, resulting in suboptimal results.

In addition, while MTL is a solution, a significant challenge lies in the potential interference among tasks, which may disturb parameter updates and degrade overall performance. The Segment Anything Model (SAM) [14] is a large-scale vision model trained on extensive datasets, offering robust feature extraction capabilities. Leveraging its pre-trained encoder to extract shared features can effectively address this issue. However, SAM's training data is primarily natural images, and its applicability to RSI remains underexplored. Furthermore, SAM is based on Transformer architecture, which relies on global attention mechanisms. Given that RSI contains numerous small-scale objects, there is a need for the network to possess strong local attention capabilities.

Inspired by recent advancements in MTL and SAM, we propose HEASSNet, an MTL network designed to achieve HE and SS from optical RSI. HEASSNet avoids relying on specific optimization strategies for HE and SS, choosing a simple architecture with reduced complexity. Specifically, we use the pre-trained encoder of SAM and propose a LoRA fine-tuning method with convolution to address the local limitation of

Yachen Wang, Yejian Zhou and Huayong Tang are with College of Information Engineering, Zhejiang University of Technology, Hangzhou, 310023, P.R.China.

Guanyong Wang is with the School of Information and Science, North China University of Technology, Beijing 100144, China.

Lei Zhang is with the School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen 518107, China.

The paper is supported by National Natural Science Foundation of China (Grant No. 62471438 and 62401018), the Zhejiang Provincial Natural Science Foundation of China (Grant No. LY23F010012). (Corresponding author: Yejian Zhou, email: yjzhou25@zjut.edu.cn;)

SAM. HE and SS are executed in independent branches. The SS branch uses the swin transformer (SwinT), which excels at capturing long-range dependencies and ensuring accurate judgments in detail and overall. The HE branch consists of convolutional blocks, which effectively obtain details and make accurate predictions but are limited in capturing global features. For this, we propose a feature enhancement and aggregation gate (FEA-Gate). The module enhances local and global features, selectively aggregating this information into the HE branch, improving its global perception and contextual information. The height result is input into the prompt encoder and is summed pixel-wise with the features of the SS branch, effectively distinguishing confused categories and improving segmentation accuracy.

In summary, our contributions are summarised as follows:

- We propose HEASSNet, an MTL network designed to interpret high-resolution RSI through SS and HE. HEASSNet leverages joint learning to constrain and optimize performance, achieving state-of-the-art results on diverse datasets.
- We propose a fine-tuned method based on LoRA. Using convolution operations for multi-scale features in LoRA, solving the limitations of the encoder and enhancing the local prior, while retaining the strong generalization of SAM, making it more suitable for RSI.
- To achieve a positive transfer of favorable information in cross-task, we propose the FEA-Gate. The module effectively filters out irrelevant and redundant features from various modalities, allowing the incorporation of sufficiently complementary information to form a comprehensive representation for predicting height value.

Organization. In Section II, we review the existing work closely related to ours. We propose the details of HEASSNet Section III. The discussion of experiments is given in Section IV. In Section V, we conclude our paper.

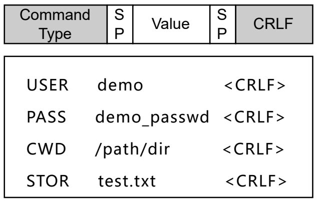
II. Background and Motivation

In this section, we introduce the primary technical concepts of protocol fuzzing and clarify the main challenges we aim to address in this paper.

A. Protocol Fuzzing

To ensure effective and reliable information sharing on the Internet, the Internet Engineering Task Force (IETF) and published as Requests for Comments (RFCs). For example, the File Transfer Protocol (FTP) is based on RFC 959. These protocols outline the general structure and sequence of message exchanges. As shown in Fig. 1, an FTP message consists of a message command type, key-value pairs, and carriage return and line feed characters (CRLF). The required sequence of FTP messages is depicted in Fig. 2: the protocol implementation begins in the INIT state to the AUTH state upon receiving USER and PASS-type messages. From the INIT state to the TRAN state, the protocol must receive at least one more specific type and structure of message besides the USER and PASS messages.

Fuzzing tools automatically generate message sequences and send them to the protocol implementation. Ideally, these message sequences should adhere to the required structure and order of the protocol.



```
USER   demo          <CRLF>
PASS   demo_passwd   <CRLF>
CWD    /path/dir     <CRLF>
STOR   test.txt      <CRLF>
```

Fig. 1. FTP command structure and an example of FTP request from Lightftp.

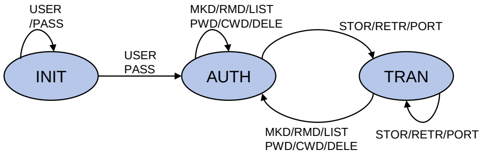


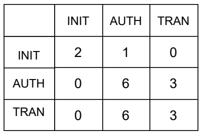
Fig. 2. FTP state model.

B. Task Scheduling

Hierarchical Approaches decompose the scheduling process into two discrete phases: (1) **Inter-state scheduling:** This phase involves selecting a state to fuzz using a state scheduling algorithm based on the priority or relevance of the state. (2) **Intra-state scheduling:** Once the target state is selected, a general scheduling algorithm is applied to optimize fuzzing within that state. By separating these phases, hierarchical approaches allow for more nuanced control over the fuzzing process. For example, if we want to test a protocol after a handshake is completed, the inter-state scheduling phase will first prioritize this state. Then, the intra-state scheduling phase will generate specific test cases to be executed within the post-handshake state using traditional scheduling methods (e.g., seed scheduling, byte scheduling, mutation strategy scheduling). In this paradigm, the heuristics used by the scheduling process mainly fall into three categories, namely rarity-preferred, performance-preferred, and complexity-preferred.

Rarity-preferred heuristics allocate more resources to seldomly exercised states, hypothesizing that these states harbor more undiscovered adjacent states or code logics [19, 110, 123, 162]. Performance-preferred heuristics prioritize states demonstrating higher code coverage or bug discovery rates [30, 55, 56, 110, 123]. Furthermore, some works utilize complexity-preferred heuristics, favoring states with greater complexity (i.e., connected to more basic blocks) or deeper states (i.e., further from the initial state) [45, 57]. For example, ICS3Fuzzer [45] inclines to choose the deeper states and those states that exercise more basic blocks. As a generation-based fuzzer, Pulsar [57] calculates the weight of all states that can be reached from the current state and then selects the state that has the maximum weight to be tested next. In detail, the weight of a state is calculated as the sum of all mutable fields in a fixed number of transitions.

However, since all these state selection algorithms are implemented and evaluated separately on different platforms and targets, it is difficult to make a fair comparison and achieve conclusive findings. Liu et al [85] evaluate the three existing state selection algorithms of AFLNet [123], including a rarity-preferred algorithm, an algorithm that randomly selects states, and a sequential state selection algorithm. They find that these algorithms achieved very similar results in terms of code coverage. They attribute the reasons to the coarse-grained state abstraction of AFLNET and the inaccurate estimation of the state productivity. Therefore, they propose the AFLNETLEGION algorithm [85] to address these issues, which is based on a variant of the Monte Carlo tree search algorithm [84].



	INIT	AUTH	TRAN
INIT	2	1	0
AUTH	0	6	3
TRAN	0	6	3

Fig. 3. FTP state transition out-degree and in-degree matrix diagram.

C. Motivation

Existing fuzzing strategies for stateful protocol implementations often exhibit critical shortcomings: (1) **Short-sighted fuzzing strategies hinder the exploration of deep program state paths.** Conventional fuzzers, such as AFL-based tools, prioritize minimizing message interactions by selecting the shortest viable test cases to transition between protocol states. While this approach optimizes resource consumption, it overlooks the nuanced impact of intermediate state transitions on subsequent fuzzing campaigns. For instance, consider a FTP implementation where the TRAN state can be reached via multiple command sequences—e.g., [PASS, USER, STOR], [PASS, USER, LIST, STOR], or the longer [PASS, USER, CWD, LIST, MKD, STOR]. Although the latter incurs higher fuzzing overhead due to additional commands (e.g., MKD creating a directory), these operations fundamentally alter the server's state, enabling richer feedback for subsequent inputs. However, most fuzzers favor shorter sequences, discarding semantically deeper paths that could expose subtle vulnerabilities. Moreover, state transitions in protocol implementations exhibit varying degrees of complexity. As illustrated in Fig. 3, transitions from AUTH to TRAN involve significantly more paths than those from INIT to AUTH, suggesting that state coverage alone is insufficient for effective fuzzing. By neglecting longer, more intricate command sequences, current tools fail to exercise critical edge cases, ultimately limiting their ability to uncover deep-state vulnerabilities. This work addresses this gap by proposing a state-aware fuzzing strategy that systematically prioritizes path diversity over mere state coverage.

(2) **Existing grey-box fuzzing approaches for stateful network protocols exhibit significant limitations in their evaluation methodology.** While recognizing the importance of state selection and providing heuristic algorithms like RANDOM, ROUND-ROBIN, and FAVOR, current methods suffer from oversimplified assessment criteria that fail to accurately evaluate the potential of state sequences. The predominant approach narrowly focuses on bug discovery rate for state selection while ignoring execution efficiency, and prioritizes shortest-path sequences over more exploratory alternatives. To address these shortcomings, this study proposes a dual-objective optimization framework that simultaneously considers both execution time and bug discovery rate. Consequently, the current evaluation methodology exhibits limited capability in precisely quantifying the intrinsic value of discrete protocol states and their associated transition sequences, thereby adversely affecting the fuzzer's overall testing efficacy.

III. Method

In this section, we first model the system using a finite-state machine and define our optimization objectives. Subsequently, we outline the details of the key components, including .

A. Problem Definition

During the fuzzing of protocols, each state transition may correspond to the occurrence of an event. Finite State Machines (FSMs) describe the potential behavior of a system through states and the transitions between them, making FSMs particularly suitable for modeling event-driven systems. By using FSMs to model protocol implementations, it is possible to construct mappings between inputs and protocol state transitions. This mapping aids fuzzing tools in generating inputs to check the security of deep program states and their transitions. The main parameters used in this section are listed in Table I.

A Mealy FSM is a six-tuple :

$$\mathcal{A} = (Q, q_0, I, \Lambda, \delta, \lambda) \tag{1}$$

where $Q = \{q_1, q_2, \ldots, q_n\}$ is a finite set of states, q_0 represents the initial state of the system, $I = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ is the input alphabet, and $\Lambda = \{o_1, o_2, \ldots, o_n\}$ is the output alphabet; $\delta : Q \times I \rightarrow Q$ denotes the state transition function and can be expressed as $\delta = \{(q, \sigma, q') : \delta(q, \sigma) = q'\}$. $\lambda : Q \times I \longrightarrow \Lambda$ represents the set of potential outputs or observations for this transition.

Meanwhile, let I^* (excluding ϵ) represents the set of finite-length sequences on I, and let Λ^* represents the state transition outputs over Λ. In this regard, by successive iterations, for any $q_0 \in Q$, δ can be extended over a k-length string $s_k = \sigma_{i_1}\sigma_{i_2}\ldots\sigma_{i_k} \in I^*$ by $\delta(q_0, s_k) := \delta(\delta(\ldots\delta(\delta(q_0, \sigma_{i_1}), \sigma_{i_2})\ldots), \sigma_{i_k})$. Also the output function λ can be extended over a k-length string $p_k = o_{j_1}o_{j_2}\ldots o_{j_k} \in \Lambda^*$ by $p_k = \lambda(q_0, s_k) := \lambda(\lambda(\ldots\lambda(\lambda(q_0, \sigma_{i_1}), \sigma_{i_2})\ldots), \sigma_{i_k})$.

Therefore, given a Mealy FSM $\mathcal{A}$, there exist different input-output pairs (Δ_i, O_j) composed of sequences $(\sigma_{i_1}, o_{j_1})(\sigma_{i_2}, o_{j_2})\ldots(\sigma_{i_k}, o_{j_k})$, where the state transition function $\delta(q_0, s_k)$ and the output function $\lambda(q_0, s_k)$ are defined over different existing sequences of events $s_k = \sigma_{i_1}\sigma_{i_2}\ldots\sigma_{i_k}$, ensuring that $|\delta(q_0, s_k)| > 0$ and $|\lambda(q_0, s_k)| > 0$, respectively.

For any protocol implementation, a corresponding Mealy FSM $\mathcal{A}$ can be constructed. During each round of fuzzing,

TABLE I
MAIN PARAMETERS

Parameter	Sign	Content
Finite Set of States	Q	$q_1, q_2, \ldots, q_n$
Initial State of The System	q_0	/
Input Alphabet	I	$\sigma_1, \sigma_2, \ldots, \sigma_m$
Output Alphabet	Λ	$o_1, o_2, \ldots, o_n$
State Transition Function	δ	$\delta(q, \sigma) = q'$
Set of Outputs for State Transition	λ	$Q \times I \to \Lambda$
State-sequence Instance Pairs	C	$C_1, C_2, \ldots, C_l$

the fuzzing tool $\mathcal{F}$ selects appropriate *state-sequence instance pairs* $C = (q_i, s_k^j)$ for testing, where $q_i \in Q$ represents the target state to be tested in the current round. Let m_q denote the number of seed sequences (non-fixed and state-dependent) that can reach q_i, from which one seed sequence s_k^j is selected for testing.

Formally, if the protocol implementation triggers state transitions C resulting in a crash M, the crash type is recorded. This process iterates until reaching the maximum execution time limit $T_{\max}$. The experimental objective is to maximize the number of M within the limited T.

B. Optimization Goal

In fuzzing, we aim to maximize the number of discovered crashes (M) while maintaining execution time below the predefined threshold $T_{\max}$. Our formulation for the optimization objective reads:

$$\begin{aligned} &\arg\max M \\ &\text{s.t.} \quad T \leq T_{\max} \end{aligned} \tag{2}$$

First, define the non-negative integer variable x_{q,s_k} to represent the number of times the state sequence pair (q, s_k) is selected, where q is a decision option from the candidate choice set Q, and s_k is a seed sequence from the option q's corresponding seed sequence set m_q, with the constraint $x_{q,s_k} \in \mathbb{Z}^+$.

$$x_{q,s_k} \in \mathbb{Z}^+ \quad \forall q \in Q,\ s_k \in m_q \tag{3}$$

Crash Model. In this section, we aim to model the crashes M obtained through fuzzing. We define M as the cumulative crashes across all rounds generated by each state-sequence instance pair C in the fuzzing process:

$$M = \sum_{q=1}^{n} \sum_{s_k \in m_q} \sum_{p=1}^{x_{q,s_k}} \left(M_E^{q,s_k}(p) + M_N^{q,s_k}(p) \right) \tag{4}$$

where M_E^{q,s_k} denotes crashes caused by protocol violations, M_N^{q,s_k} represents crashes during specification-compliant execution.

Execution Time Model. We define the execution time T during testing as the sum of execution times for each state-sequence instance pair C:

$$T = \sum_{q=1}^{n} \sum_{s_k \in m_q} \sum_{p=1}^{x_{q,s_k}} t^{q,s_k}(p) \tag{5}$$

where t^{q,s_k} denotes the total execution time summed across all rounds for each state-sequence instance pair.

Value Model. To accurately evaluate the exploration potential of each state sequence instance pair and ensure the selection of higher-value pairs in subsequent fuzz testing, thereby maximizing crash discovery efficiency within limited execution time, we propose a novel value assessment model E^{q,s_k}. This model quantitatively estimates the exploration value (encompassing both crash count and execution time) of instance pairs to optimize testing performance:

$$E^{q,s_k} = \sum_{p=1}^{x_{q,s_k}} E^{q,s_k}(p) \tag{6}$$

where E^{q,s_k} denotes the total exploration value summed across all $selected\_time$ experimental rounds for each state-sequence instance pair C, calculated as the accumulation of $E^{q,s_k}(p)$ values obtained in each round.The exploration value per experiment $E^{q,s_k}(p)$ is defined as:

$$E^{q,s_k}(p) = F^{q,s_k}(p) \cdot T^{q,s_k}(p) \tag{7}$$

$$F^{q,s_k}(p) = 2^{\log(\text{paths\_discovered(p)}+\alpha(p) \cdot M_E^{q,s_k}(p)+M_N^{q,s_k}(p)+1)} \tag{8}$$

$$\alpha(p) = \frac{M_E^{q,s_k}(p)}{M_E^{q,s_k}(p) + M_N^{q,s_k}(p) + \epsilon} \tag{9}$$

The proposed state value evaluation formula consists of two components. **The first component is the crash discovery value, as formulated in Equation (8).** This metric is adapted from AFLNet's state value estimation formula, with modifications introduced to optimize crash count maximization under our testing framework. This component integrates the following key factors: (1) code path coverage ($paths\_discovered$), where greater coverage indicates higher probability of discovering potential vulnerabilities; (2) protocol-violation crashes (M_E^{q,s_k}), whose impact is reduced through a weighting coefficient $\alpha(p)$ since such crashes typically cannot trigger genuine vulnerabilities; $\alpha(p)$ denotes the proportion of M_E^{q,s_k} among the total number of crashes and ϵ is a smoothing factor (ϵ=1) to prevent division by zero; and (3) valid protocol crashes (M_N^{q,s_k}), which exhibit higher vulnerability detection value as they conform to protocol specifications while triggering abnormal execution. The logarithmic relationship in this factor demonstrates a negative regulatory effect on exploration value as testing iterations increase, thereby effectively preventing excessive retesting of low-efficiency regions and improving the overall efficiency of fuzz testing.

$$T^{q,s_k}(p) = \beta(p) \cdot \frac{1}{2^{\log_{10}(\log_{10}(t^{q,s_k}(p)+\epsilon))}} \tag{10}$$

$$\beta(p) = \frac{50}{\epsilon + 1500 - \frac{1500}{1+e^{-0.15(T(p)-10)}}} \tag{11}$$

The second part quantifies execution efficiency of state sequence instance pairs, as formulated in Equation (10).

Experimental studies demonstrate that the path coverage and crash discovery efficiency during fuzzing exhibit significant time-dependent features, manifested by a notably higher discovery rate per unit time in the initial testing phase compared to later stages. To accurately characterize this dynamic behavior, this study proposes a time-weighted dynamic reward mechanism based on experimental data. By introducing a time gain function to provide progressive value compensation for crashes and paths discovered in later stages, the mechanism effectively addresses the issue of diminishing testing returns over time. $\beta(p)$ is empirically derived through extensive experimental data fitting, where $T(p)$ represents the cumulative execution time of the ongoing fuzz testing process, and ϵ is a minimal positive constant (typically set at machine epsilon level) incorporated to prevent division by zero and maintain numerical stability in the computation.

Meanwhile, to optimize the execution time of individual test cases, the model incorporates a time penalty factor with negative derivative properties (corresponding to the right-hand side of the equation). This design is theoretically grounded in two fundamental observations: prolonged execution time linearly reduces testing throughput, while execution efficiency demonstrates significant positive correlation with potential exploration value. The dual-regulation mechanism, integrating time-gain rewards and execution-time penalties, enables optimal resource allocation across both temporal and test-case dimensions through dynamic value assessment. This approach not only adheres to the fundamental exploration-exploitation trade-off principle in fuzz testing but also adaptively prioritizes high-potential test cases, thereby significantly improving overall testing efficiency. Here, $t^{q,s_k}(p)$ denotes the execution time of the current test cycle, whose value is determined based on the theoretical framework of AFLNet's state evaluation formula and subsequently validated through experimental studies.

C. MAB

This study employs the Multi-Armed Bandit (MAB) algorithm as the core optimization strategy, which dynamically balances the exploration-exploitation trade-off to select the optimal combination from candidate test case state sequences for maximizing the objective function. The algorithm effectively enhances optimization efficiency while ensuring solution quality. The theoretical foundations of the MAB algorithm and its applicability to this research will be elaborated in the following section.

Exploration vs. Exploitation. The trade-off between exploration and exploitation is an important concept in game theory. Inspired by a player's choice to maximize the reward a slot machine, the MAB model is proposed to solve this problem. According to the definition, a classic MAB model contains N parallel arms, and only one arm is selected each time. The expectation of reward for arm i ($i \in \{1, 2, \ldots, N\}$) is defined as R_i. The key to the MAB problem is to maximize the total reward within finite arm selections.However, before trying a certain arm, its reward is unknown. The process of **exploration** performs trials on an arm to acquire a more accurate calculation of its reward. When the rewards of all the arms are known, choosing the arm with the maximum reward is the process of **exploitation**.In conclusion, choosing the best arm (exploitation) will maximize the current total reward. In the long term, making trials on reward-unknown arms (exploration) will help reach a larger total reward. It is our goal to weigh the strengths and weaknesses between exploration and exploitation in the MAB problem.

In the experiment, the system comprises n states, where each state q is associated with a set of reachable seed sequences m_q.

$$a_{i,j} = C_l \in C \tag{12}$$

$$N = \sum_{q \in Q} |m_q| \tag{13}$$

Each state-seed sequence pair represents a fuzzing operation unit, i.e., a bandit arm $a_{i,j}$, resulting in a total of N distinct bandit arms, where N is the sum of reachable seed sequences across all states.

The reward $R(C_l, n_l)$ of each bandit arm $a_{i,j}$ is defined as E^{q,s_k}, representing its expected fuzzing effectiveness. In the algorithm design, the parameter n_l represents the cumulative selection count of state sequence pair C_l (where $n_l \equiv x_{q,s_k}$, with x_{q,s_k} denoting the original counting variable). For notational simplicity, we consistently adopt n_l as the standardized representation throughout this paper.

$$R(C_l, n_l) = E^{q,s_k} = \sum_{p=1}^{x_{q,s_k}} E^{q,s_k}(p) \tag{14}$$

Next, we can calculate a final score for the combinations to make decisions. UCB1 [27] is a classic answer to the MAB problem, and we calculate scores based on it as

$$\begin{aligned} \text{Score}(C_l, n_l) &= \overline{R}(C_l, n_l) + U(C_l, n_l) \\ &= \frac{\sum_{p=0}^{n_l} R(C_l, p)}{n_l} + \gamma \cdot \sqrt{\frac{\ln\left(\sum_{C_l \in C} n_l\right)}{n_l}} \end{aligned} \tag{15}$$

In the fuzzing framework, **C** denotes the set of all instance pairs. The scoring function $\text{Score}(C_l, n_l)$ consists of two components:

where:$R(C_l, n_l)$ represents the average reward of pair C_l over n_l previous rounds (reflecting historical performance for exploitation) $U(C_l, n_l)$ is the upper confidence bound component. The algorithm begins with a **pioneering phase** that enforces $\forall C_l \in C, n_l = 1$. In subsequent rounds, the system dynamically selects:

$$C_{\text{next}} = \underset{C_i \in C}{\text{argmax}}\ \text{Score}(C_l, n_l) \tag{16}$$

The hyperparameter γ controls the exploration-exploitation trade-off.

D. Algorithm Design

Algorithm 1 The Fuzzobj algorithm

```
Input  : $P_t$: Protocol Implementation
         $s_{k0}$: Initial Seed Queue with state $q_0$
         $d$: Dictionary Corpus
         $D$: Current fuzzing depth
Output: $M$: Crash sequence count
1  Initialize FSM $S$ and Seed Queue $s_k 0$ with $q_0$;
2  Pioneer Phase: Each $C_l$ is selected once to initialize all $n_l \leftarrow 1$; After each fuzzing iteration, compute $SWVA\text{-}UCB(C_l, n_l)$ to guide future selection;
3  repeat
4      State-sequence Pair Selection Phase:
       foreach $C_l \in C$ do
5          SWVA-UCB$(C_l, n_l) \leftarrow$ ComputeScore$(C_l, n_l)$
6      $\mathcal{K} \leftarrow$ GetTopKPairs$(C, K)$ ; // Rank by SWVA-UCB score
7      $C_l^* \leftarrow$ BoltzmannSelect$(\mathcal{K})$
8      Energy Allocation Phase:
       $E \leftarrow$ AssignEnergy$(C_l^*)$
9      Mutation Testing Phase:
       for $i \leftarrow 1$ to $E$ do
10         $INIT \leftarrow$ CalculateDepth$(D, R)$
11         if $NotINIT$ then
12             $s_k' \leftarrow$ StateGuidedMutate$(s_k, d, q)$
13         else
14             $s_k' \leftarrow$ RandomMutate$(s_k, q)$
15         if $ContextAware(s_k', d)$ then
16             $M_E \leftarrow M_E \cup \{s_k'\}$
17         $R' \leftarrow$ SendToServer$(P_t, s_k')$
18         if $IsCrash(s_k', P_t)$ then
19             $M_N \leftarrow M_N \cup \{s_k'\}$
20         else if $IsInteresting(s_k', P_t, q)$ then
21             $m_q \leftarrow m_q \cup \{s_k'\}$
22             $Q \leftarrow$ UpdateFSM$(Q, R')$
23 until timeout $T$ or abort-signal;
24 return $M = M_E + M_N$
```

1) Overview: To enhance algorithmic dynamism and mitigate errors induced by fuzzing randomness, we propose Fuzzobj, an optimized variant of UAB1, with the following workflow: The proposed state-guided protocol fuzzing framework takes four key inputs: the target protocol implementation Pt, an initial sequence s_{k0} containing the starting state q_0, dictionary corpus d, and the fuzzing depth D. The primary output is the total number of crashes M detected during testing.

The fuzzing process begins with an initialization and pioneer phase (Lines 1-2), where all state-sequence pairs undergo preliminary testing. During the state-sequence pair selection phase, the fuzzer first computes the SWVA-UCB score for each state-sequence pair (Lines 4-5), then constructs a Top-K candidate pool based on these scores (Line 6). The BoltzmannSelect algorithm is subsequently employed to select the target state-sequence pair Cl from this pool (Line 7), effectively guiding the fuzzer's exploration of the target server. The energy allocation phase dynamically distributes fuzzing resources according to historical testing results and selection frequencies (Line 8). In the mutation testing phase, the framework applies different fuzzing strategies to mutate sequence s_{k0} based on both the protocol's program state and current fuzzing depth (Lines 10-14). Following mutation, each sequence undergoes validity verification through a content-aware module (Lines 15-16). Sequences causing protocol specification violations are recorded as ME (protocol errors) and immediately proceed to the next iteration. Valid sequences are sent to Pt for execution (Line 17). The fuzzer maintains two categories of mutated sequences s_k' : those that trigger crashes in Pt, and those that improve code or state coverage. For the latter case, the system correspondingly updates the Finite State Machine (FSM). This iterative process continues until the allocated fuzzing resources for the current round are exhausted, at which point a new state-sequence pair is selected for subsequent testing. The final output M represents the cumulative count of valid crashes detected throughout the testing campaign.

2) Detailed Techniques of Fuzzobj: **Dynamic Optimization of Evaluation Bias.** To address the evaluation bias caused by the dynamic variations in path and crash rewards during fuzz testing, the proposed algorithm introduces a sliding window mechanism to dynamically optimize the assessment of state-action pairs. This mechanism restricts the influence of historical data by retaining only the most recent W data samples, thereby enabling rapid forgetting of obsolete information and ensuring that the model more accurately captures the reward distribution characteristics of the current testing phase.

Mitigating Reward Variance in Fuzzing Evaluation. Due to the inherent stochasticity in fuzzing testing, even high-value state-sequence pairs may yield relatively low crash or path rewards, while low-value pairs could conversely produce higher rewards. Such randomness can significantly compromise the accuracy of test evaluation. To mitigate this issue, we propose a variance estimation mechanism σ^2 for state-action pairs to effectively reduce the interference caused by random fluctuations during the testing process.

The estimators for both the mean and variance are reconstructed as:

$$\hat{\mu}(C_l, n_l) = \frac{1}{\min(W, n_l)} \sum_{p=n_l-W+1}^{n_l} R(C_l, p) \tag{17}$$

$$\hat{\sigma}^2(C_l, n_l) = \frac{1}{\min(W, n_l) - 1} \sum_{p=n_l-W+1}^{n_l} (R(C_l, p) - \hat{\mu}(C_l, n_l))^2 \tag{18}$$

Adaptive Exploration-Exploitation Balance. To address the "premature search space convergence" phenomenon in fuzzing testing - where traditional UCB algorithms' deterministic selection strategy causes early-stage fixation on suboptimal test paths while neglecting continued exploration of potentially high-value regions - we propose an enhanced exploration mechanism based on the Boltzmann distribution. This innovative approach converts the sliding-window variance-aware UCB score (SWVA-UCB) into an exponentially weighted probability distribution, employing a time-dependent temperature parameter $\eta(t)$ to dynamically modulate exploration intensity. Specifically, higher initial temperature values facilitate broad exploration during early testing phases, while progressively decreasing temperatures gradually shift focus toward higher-reward actions. This relative action-value-based softmax strategy effectively combines the advantages of: (1) overcoming the local optimum limitation inherent in greedy algorithms, and (2) mitigating the indiscriminate exploration characteristic of $\epsilon - greedy$ methods, thereby achieving optimal adaptive balance between exploration and exploitation throughout the testing process.

The improved upper confidence bound (UCB) based on our modification is expressed as:

$$\text{SWVA-UCB}(C_l, n_l) = \hat{\mu}(C_l, n_l) + \sqrt{\frac{2\hat{\sigma}^2(C_l, n_l) \ln\left(\sum_{C_l \in C} n_l\right)}{n_l}} \tag{19}$$

We propose a two-phase selection strategy to optimize the exploration process: First, we select the top $K = \lceil \log N \rceil$ candidate arms based on their SWVA-UCB scores to form set $\mathcal{K}$, followed by Boltzmann selection within $\mathcal{K}$. This mechanism achieves dynamic exploration through adaptive temperature regulation, significantly improving computational efficiency while maintaining selection quality. The Boltzmann distribution probability generation, adaptive temperature function, and final Boltzmann selection are expressed by the following formulas:

$$P(C_l, n_l) = \frac{\exp\left(\eta(C_l, n_l) \cdot \text{SWVA-UCB}(C_l, n_l)\right)}{\sum_{k=1}^{N} \exp\left(\eta(C_l, n_l) \cdot \text{SWVA-UCB}(C_k, n_k)\right)} \tag{20}$$

$$\eta(C_l, n_l) = \frac{1}{\sqrt{\frac{1}{N} \sum_{k=1}^{N} \hat{\sigma}^2(C_k, n_k) + \epsilon}} \tag{21}$$

$$P(C_l, n_l) = \begin{cases} \frac{\exp(\eta(C_l,n_l)\,\text{SWVA-UCB}(C_l,n_l))}{\sum_{k\in\mathcal{K}} \exp(\eta(C_l,n_l)\,\text{SWVA-UCB}(C_k,n_k))}, & l \in \mathcal{K}, \\ 0 \qquad , & \text{otherwise.} \end{cases} \tag{22}$$

Content-aware Module. we executed the protocol implementation and conducted a series of interactions with test endpoints. We capture and analyze network packets exchanged during these interactions, focusing on fields related to protocol interaction commands, protocol states, and abnormal behaviors. Referring to the protocol's RFC documentation, we constructed mappings of protocol state transitions and interaction commands, using these mappings as input from prior knowledge for the fuzzing tool. Using these inputs, we developed a content-aware module. This module employs a filtering method to screen mutated samples. At a high level, this approach safely discards a large number of invalid samples without affecting the optimal solution. Specifically, the fuzzer drives the protocol implementation through state transitions by providing protocol commands (events) to detect potential vulnerabilities. Based on the FSM modeling results of the protocol, all interpretable protocol command types can be abstracted into an input set $\sum$. As described in Algorithm 2, when the fuzzer mutates command types of protocol within $\sum$ to generate new instances, if the mutated instance cannot be interpreted by the protocol implementation (i.e., it does not belong to $\sum$), it will not be added to the message queue Q for transmission to the target server, but instead is placed in the crash queue M_E, thus reducing the waste of fuzzing resources.

State-guided Mutation module. We designed a state transition dictionary d based on prior fuzzer knowledge. This dictionary stores message sequences that transition the protocol server from the current state A to the target state B. Additionally, the current program states S and its depth in the state chain D as parameters for the fuzzer. We then adjust the fuzzing strategy based on two scenarios: (1) if the depth of the selected fuzzing state S in the loop is less than the minimum initialization state length, and (2) if a state repeatedly appears in the state chain without causing a crash (e.g., the same state information appears at different depths in the state chain). When the fuzzing iteration encounters these situations, Fuzzobj invokes our state-guided mutation module.

IV. Experiment

In this section, we first describe the datasets used and provide execution details. We then evaluate the performance of HEASSNet in comparison to classic algorithms for HE and SS tasks, using two well-known public datasets: ISPRS Potsdam and DFC2023. Following this, we compare our method against four commonly used backbones and perform SS and CD tasks using the SECOND dataset, thereby verifying the portability of our shared encoder strategy. Finally, we conduct ablation experiments to demonstrate the contributions of each key component within HEASSNet.

A. Datasets and Execution Details

1) Datasets: To verify the effectiveness of HEASSNet, HE and SS experiments were carried out on the DFC2023 and ISPRS Potsdam datasets, and the SS and CD experiments were conducted on the SECOND dataset.

DFC2023: The DFC2023 [15] (Data Fusion Contest 23) dataset provides satellite images, digital surface models, and semantic labels of buildings in 17 cities from six continents. The DSM is derived from stereo images captured by Gaofen-7 and WorldView satellites, achieving a 2-meter ground sampling distance. The dataset contains 1773 images, each sized at 512×512 pixels, with accurately annotated semantic labels. The semantic maps outline building footprints and do not distinguish the building type. For the experiment, due to the limitation of GPU memory, all images are cropped into 7092 patches of 256×256 size. These patches are randomly divided into training, validation, and test sets, with data sample sizes of 4964, 709, and 1419 respectively.

Potsdam: The Potsdam dataset includes 38 optical RSI, digital surface models, and semantic labels. Each image boasts a high resolution of 5 cm per pixel and dimensions of 6000×6000 pixels. The dataset includes six categories:

TABLE II
QUANTITATIVE RESULTS OF SS AND HE ON THE DFC2023 DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Height Estimation Results						
Method	MAE ↓	RMSE ↓	SSIM ↑	$\delta1$ ↑	$\delta2$ ↑	$\delta3$ ↑
Im2height	3.102	7.300	85.84	51.08	55.87	62.36
PixelFormer	2.533	5.930	82.75	34.55	50.47	59.37
GLPDepth	2.511	5.939	83.52	32.56	49.38	59.12
HTCDCNet	2.479	5.337	81.35	35.21	51.66	60.57
NewCRFs	2.171	5.305	85.21	40.06	56.47	65.06
LUMNet	1.738	**4.446**	88.17	53.12	69.21	76.77
Ours	**1.705**	4.681	**89.22**	**56.67**	**72.23**	**79.27**

Semantic Segmentation Results				
Method	OA ↑	mIoU ↑	Kappa ↑	F1 ↑
Swin-Unet	88.21	71.81	65.63	82.79
SegNet	88.43	72.78	67.12	83.56
DeepLab3	92.66	81.95	79.63	89.82
DANet	93.24	82.87	80.77	90.38
UnetFormer	93.71	84.10	82.31	91.15
EncNet	94.37	85.57	84.11	92.05
Ours	**94.60**	**86.22**	**84.90**	**92.45**

impervious surfaces, buildings, low vegetation, trees, cars, and background. In the experiment, each image is cropped into several patches of size 256×256. These patches are divided into training, validation, and test sets in a ratio of 7:1:2, respectively.

SECOND: The SECOND [16] dataset is well-annotated for semantic CD. The dataset collects pairs of multi-temporal aerial images from multiple platforms and sensors, with each image being 512×512 in size and annotated at the pixel level. The difference in the truth label of each pair of images is the truth value of the CD. The dataset includes six land cover categories: unvegetated surface, trees, low plants, water, buildings, and playgrounds. In the experiment, all images are cropped into 11872 patches of 256×256 size and randomly divided into training, validation, and test sets, with data sample sizes of 8310, 1187, and 2375.

2) Training Details: The proposed method uses the publicly accessible PyTorch framework. Training epochs is a maximum of 40, with a batch size of 8 to converge. The initial learning rate is set to 1.0×10^{-4} and decays by a factor of 0.1 every 20 epochs to facilitate effective learning. All experiments are conducted using GPU to improve computational efficiency. We use the AdamW optimizer to optimize network parameters, with a beta of 0.9 and a weight decay rate of 0.01% to mitigate overfitting and enhance generalization capabilities.

3) Evaluation Metrics: We use a series of generally accepted metrics to evaluate the proposed method quantitatively.

For SS, we use overall accuracy (OA), Kappa coefficient, intersection-over-union (IoU), and F1 score to measure the performance of all networks in the experiment.

For HE, we use mean absolute error (MAE), root mean square error ($RMSE$), structural similarity ($SSIM$), and thresholded accuracy δ to measure the quality of the generated height map. The HE metrics are computed as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|H_i - \hat{H}_i\right| \tag{23}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(H_i - H_i)^2} \tag{24}$$

$$SSIM = \frac{(2\mu_H\mu_{\hat{H}} + C_1)(2\sigma_{H\hat{H}} + C_2)}{\left(\mu_H^2 + \mu_{\hat{H}}^2 + C_1\right)\left(\sigma_H^2 + \sigma_{\hat{H}}^2 + C_2\right)} \tag{25}$$

$$\delta_K = \max(\frac{H}{\hat{H}}, \frac{\hat{H}}{H}) < 1.25^K, K \in 1, 2, 3 \tag{26}$$

where H and $\hat{H}$ represent the true image and predicted height map, N is the number of all images in each epoch, μ, σ and $\sigma_{H\hat{H}}$ represent the mean, variance and covariance of H and $\hat{H}$. In addition, C_1 and C_2 are constants with small values to avoid the numerator or denominator being zero.

B. Competitors

For SS, the proposed method is benchmarked against several CNN-based networks and Transformer-based models. Specifically, they are SegNet [17], EncNet [18], DeepLabV3 [19], DANet [20], Swin-Unet [21], and UnetFormer [22]. For HE, the algorithms compared with the HEASSNet can be divided into two groups: one is specialized HE networks for RSI—Im2Height [23], LUMNet [24], and HTCDCNet [25]; the other is depth estimation networks used for similar pixel value analysis—PixelFormer [26], GLPDepth [27], and NewCRFs [28].

C. Performance Comparison

In this section, we report the quantitative and qualitative comparison results of the proposed HEASSNet with other methods on HE and SS. Then, we analyze these results to demonstrate the HEASSNet's effectiveness.

Table II reports the quantitative comparison results on the DFC2023 dataset. Compared with the above methods, the proposed HEASSNet performs better for both SS and HE. Specifically, for SS, our approach achieves OA of 94.60%, Kappa of 86.22%, mIoU of 84.90%, and an average F1 score of 92.45%. For HE, our approach achieves MAE of 1.705, RMSE of 4.681, and SSIM of 89.22, maintaining the best performance across different accuracy thresholds δ. These results indicate that our method is more effective and reliable. Notably, existing methods such as PixelFormer, GLPDepth, HTCDCNet, and NewCRFs rely on the maximum height of the scene as a boundary constraint, which introduces several

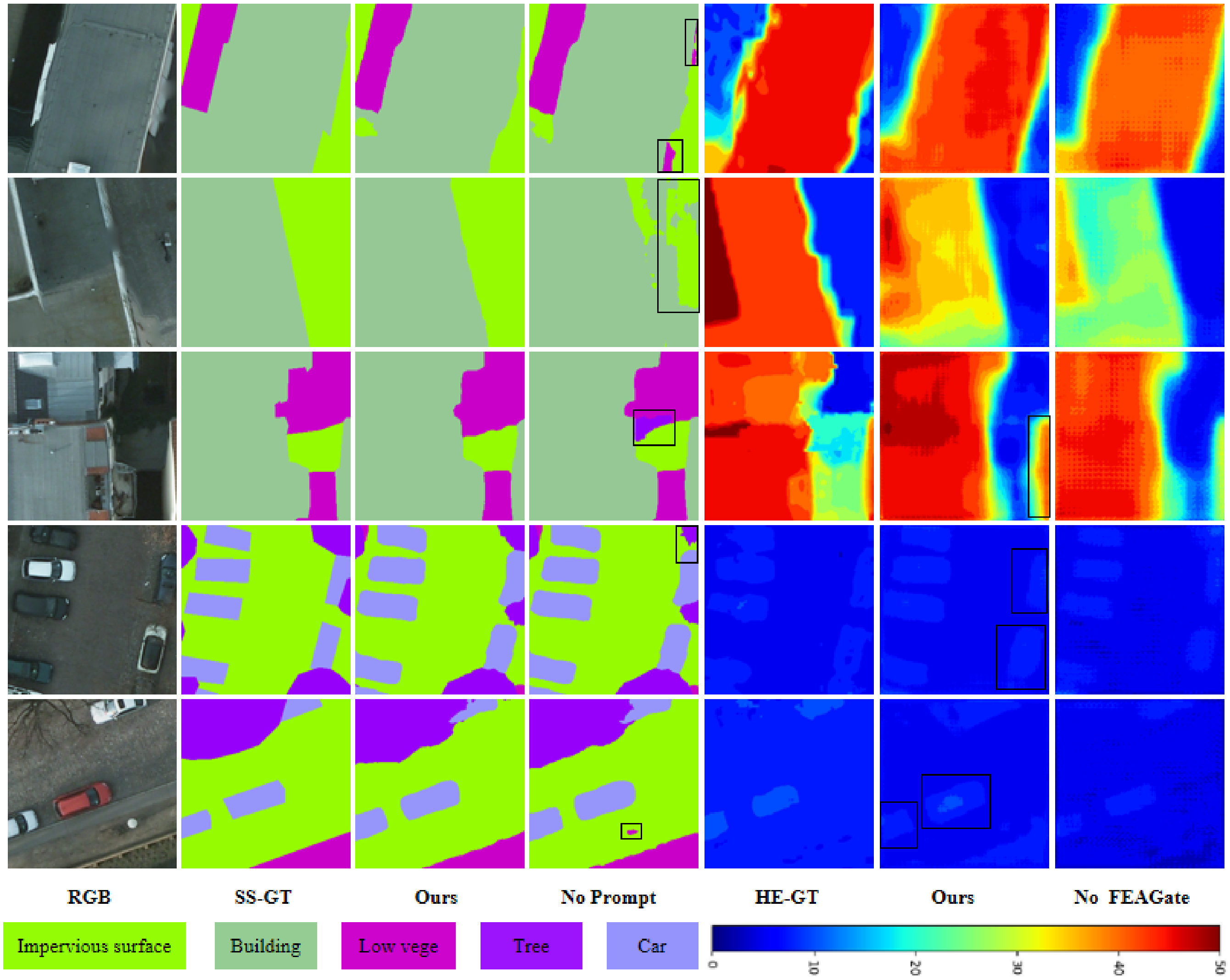


Fig. 4. Qualitative ablation results of height estimation and semantic segmentation on the Potsdam dataset. The color axis represents the corresponding height value mapping.

issues. First, due to the large proportion of background in DFC2023 datasets, it will underestimate the predicted height values. Second, the small proportion of super-tall buildings, using the maximum height value as a boundary will lead to an unreasonable increase in the overall predicted height. As a result, these methods often yield unstable and low δ values on the DFC2023 dataset, which has an unbalanced height distribution. In contrast, our methods, including Im2Height and LUMNet, generate height maps through direct regression without boundary constraints, ensuring better stability across varying height distributions.

Fig. **??** and Fig. **??** show the visualization results of the qualitative comparison. As shown in Fig. **??**, compared with the other algorithms, the predicted height values of HEASSNet for buildings within the same area are smoother and more consistent. This improvement is because of the contextual information provided by the SS branch, which helps constrain the predicted height values within the same area. Notably, in the last row of Fig. **??**, some buildings in the lower middle of the optical image are missing from the ground truth, which we believe is due to an annotation error. Despite this discrepancy, our method successfully predicts the height of these buildings and maintains consistent outlines, underscoring the reliability of our approach in practical applications. Fig. **??** presents the visualization results of SS. Compared to other segmentation algorithms, our method produces more visible boundaries, particularly at the edges of the image. This improvement is due to the height prompts generated by the HE branch, which effectively distinguishes between background and buildings.

Beyond the DFC2023 dataset, we report quantitative comparison results on the ISPRS Potsdam dataset, which has more varied, complex, and challenging scenes. As shown in Table III, our method outperforms the compared algorithms across most metrics, similar to the results observed with the DFC2023 dataset. It should be noted that the Potsdam dataset has well-balanced distribution of height values, which allows methods such as PixelFormer, GLPDepth, HTCDCNet, and NewCRFs to perform stably in terms of the δ metric. These methods do not encounter the same issues as in the DFC2023 dataset, i.e., an unbalanced height distribution will lead to less

TABLE III
QUANTITATIVE RESULTS OF SS AND HE ON THE POTSDAM DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Height Estimation Results						
Method	MAE ↓	RMSE ↓	SSIM ↑	$\delta1$ ↑	$\delta2$ ↑	$\delta3$ ↑
Im2height	3.499	6.187	97.42	91.61	98.29	98.94
PixelFormer	4.379	5.814	98.26	89.36	99.08	99.83
GLPDepth	3.795	5.655	98.24	89.62	99.08	99.85
HTCDCNet	3.033	4.454	98.61	94.42	99.46	99.96
NewCRFs	3.221	4.815	98.59	92.27	99.37	99.94
LUMNet	2.557	3.931	98.81	**96.44**	99.59	99.93
Ours	**2.304**	**3.648**	**98.93**	96.22	**99.67**	**99.98**

Semantic Segmentation Results				
Method	OA ↑	mIoU ↑	Kappa ↑	F1 ↑
Swin-Unet	80.17	64.15	73.02	77.32
SegNet	85.48	70.82	79.88	82.18
DeepLab3	93.63	83.80	91.27	90.98
DANet	89.31	77.69	85.18	87.14
UnetFormer	90.87	77.52	87.41	86.59
EncNet	92.35	78.14	89.42	86.48
Ours	**95.83**	**88.39**	**94.18**	**93.68**

stable and lower δ values.

D. The Shared Encoder Test

As mentioned earlier, we do not specify a specific optimization strategy for SS and HE, and expect to complete task transfer at the lowest cost. Therefore, in this section, we compare with other backbones to verify the extensibility of the proposed shared encoder strategy. Specifically, we conducted SS and CD tests on the SECOND dataset.

Given that the primary focus of this paper is not on CD, we did not design a specific branch for this task. Therefore, we replaced the HE branch with the decoder from ScanNet [29], demonstrating that our shared encoder strategy can achieve optimal performance with minimal migration effort. For SS, we selected DeepLabV3 and EncNet as baselines, which performed well on the previously discussed datasets. For CD, the branch of the other backbones remained consistent with our method. Table IV depicts the quantitative evaluation results of SS and CD. Our method outperforms other approaches in both tasks, achieving the best performance across most metrics. Specifically, for CD, our model achieved an OA of 87.94%, Precision of 82.68%, Recall of 75.67%, average F1 score of 78.40%, and mIoU of 66.80%. Fig. **??** provides additional visualization results of CD using our method on the SECOND dataset, further demonstrating its efficacy.

E. Ablation Analysis

In this section, we conducted ablation experiments to verify the role of key components on the Potsdam dataset and analyze the LoRA fine-tuning strategy on the DFC2023 dataset.

Key Components: Table V shows the quantitative comparison results of the ablation experiments. In the table, "Add" refers to using pixel summation to introduce global information from the SS branch into the HE branch. The results in the second and third rows demonstrate that the long-range dependencies captured by the SS branch significantly enhance the accuracy of HE. This improvement occurs because the global information provides a complete target area, constrains all pixels of the target to a specific range, and predicts the height values of each complete target smoothly and continuously. Compared to the "Add" operation, the FEA-Gate can better explore the potential of global information in the HE branch. The "Prompt" strategy involves using the predicted height map as prior knowledge for SS, effectively distinguishing boundaries between different targets and thereby improving the accuracy of SS. Fig. 4 presents the visualization results of the ablation experiments. The height prompt results incorporated in the SS branch achieve clearer and more precise boundaries. Additionally, the global features from the SS branch lead to a smoother and more consistent height map in the HE branch, with more complete outlines for each target.

The detailed results of SS are depicted in Table VI. In optical RSI, impervious surfaces do not contain color information. Height prompts can effectively distinguish this category, leading to improved results. Buildings and trees are usually taller and may exhibit shadow noise in 2D images because of the imaging mechanism. The height prompts are valuable for these objects, as they help reduce shadow noise and significantly enhance performance. Conversely, cars and low vegetation tend to have similar heights in the height prompt. Given that the dataset contains a large amount of low vegetation, this similarity of height values makes it more challenging to identify cars.

LoRA Fine-Tuning: We report the ablation results of the fine-tuning strategy in Table VII. The "Base" refers to the result of adjusting all parameters after importing the pre-trained weights. As anticipated, this approach does not yield optimal results. Adjusting all parameters weakens SAM's original feature extraction capabilities. Moreover, remote sensing datasets often lack sufficient training data to optimize the Vision Transformer model plenty. The subsequent results highlight the performance improvements achieved through LoRA fine-tuning. Notably, The initial fine-tuning of Query (Q) and Value (V) in the attention layer significantly improves the performance of HEASSNet. SAM's training data primarily consists of natural images, which differ from RSI. Therefore, customizing the fine-tuning of Q, and V parameters enables SAM to extract more meaningful features from RSI, benefiting HE and SS tasks. In Table VII, N represents the scaling factor of the feature map of LoRA within each four-layer transformer. When these multi-scale features align more closely with the traditional encoder structure, the convolutional layer's role becomes more pronounced. The strategy will lead to improved

TABLE IV
QUANTITATIVE RESULTS OF SS AND CD ON THE SECOND DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Method	MAE ↓	RMSE ↓	SSIM ↑	δ1 ↑	δ2 ↑	δ3 ↑	OA ↑	mIoU ↑	Kappa ↑
DeepLabv3	68.21	39.29	66.26	54.52	-	-	-	-	-
EncNet	68.69	38.78	66.78	53.57	-	-	-	-	-
ScanNet	-	-	-	-	87.85	82.55	75.46	78.21	66.58
ConvNext	65.41	34.60	63.25	48.90	87.33	81.16	75.20	77.59	65.80
VGG16	68.69	38.42	66.68	53.05	87.02	79.93	76.10	77.76	65.90
ResNet50	68.90	37.77	66.88	52.19	87.23	80.23	**76.70**	78.25	66.46
Swin-Trans	69.08	39.91	67.19	55.10	87.48	82.14	74.30	77.22	65.47
Ours	**69.32**	**40.13**	**67.37**	**55.48**	**87.94**	**82.68**	75.67	**78.40**	**66.80**

TABLE V
ABLATION STUDY OF THE PROPOSED MODEL ON THE POTSDAM DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Method	MAE ↓	RMSE ↓	SSIM ↑	δ1 ↑	δ2 ↑	δ3 ↑	OA ↑	mIoU ↑	Kappa ↑	F1 ↑
Base	2.658	4.193	98.77	94.33	99.52	99.93	93.77	84.18	91.43	91.21
Base+Add	2.615	4.153	98.75	94.76	99.46	99.95	94.89	86.83	92.93	92.84
Base+FEAGate	**2.283**	3.676	98.91	95.93	99.65	**99.98**	95.57	87.66	93.81	93.21
Base+FEA+Prompt	2.304	**3.648**	**98.93**	**96.22**	**99.67**	**99.98**	**95.83**	**88.39**	**94.18**	**93.68**

TABLE VI
DETAILED QUANTITATIVE RESULTS OF SS IN ABLATION EXPERIMENTS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Method	OA ↑	Kappa ↑	mIoU ↑	F1 ↑	IoU				
					impervious surfaces	building	low vege	tree	car
Base	93.77	84.18	91.43	91.21	88.29	89.36	91.92	70.47	80.87
Base+add	94.89	86.83	92.93	92.84	89.39	91.10	93.90	**78.02**	81.77
Base+FEAGate	95.57	87.66	93.81	93.21	91.43	92.81	94.26	72.16	**87.64**
Base+FEA+Prompt	**95.83**	**88.39**	**94.18**	**93.68**	**91.64**	**93.63**	**94.52**	74.82	87.35

performance in the HE branch with CNNs as the backbone, while it has a negative impact on the SS branch. Nevertheless, given the lightweight of these convolutions, the performance degradation in the SS branch remains within acceptable limits.

F. Discussion

The experimental results presented above fully demonstrate the feasibility, effectiveness, and scalability of the HEASSNet. However, as illustrated in Fig. **??** and Fig. **??**, our method and other algorithms exhibit notable errors when dealing with small and dense targets, which remains a significant challenge in the intelligent interpretation of RSI. Moreover, when height maps as prompts, the height vector is applied as an offset and added pixel by pixel to the vector from the SS branch. This approach may not fully explore the potential of the height map in SS. Future research should aim to fully make these two modalities complementary, which continues to be an area of active interest. Lastly, the development of multimodal datasets that support additional tasks (three or four) could further explore the performance of HEASSNet. We believe that HEASSNet can continue to achieve promising results.

V. CONCLUSION

In this paper, we propose HEASSNet, a novel MTL framework designed to perform HE and SS using monocular optical RSI. Rather than prioritizing specific optimization strategies for HE and SS, we focus on simplifying the network structure and reducing design complexity. Specifically, HEASSNet adopts the pre-trained encoder of SAM, and we propose a novel LoRA fine-tuning strategy to make it more suitable for optical RSI. We design two independent branches to handle the distinct attributes of HE and SS tasks. To explore the

TABLE VII
QUANTITATIVE RESULTS OF DIFFERENT BACKBONES ON THE DFC2023 DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Method	MAE ↓	RMSE ↓	SSIM ↑	$\delta 1$ ↑	$\delta 2$ ↑	$\delta 3$ ↑	OA ↑	mIoU ↑	Kappa ↑	F1 ↑
Base	1.932	4.938	88.07	52.50	70.72	78.75	94.21	85.27	83.75	91.87
Q+V	1.753	4.615	88.90	52.57	69.20	76.87	94.61	86.26	84.95	92.47
Q+V+Conv(N=1,1,1)	1.723	4.630	89.13	54.64	70.56	77.63	**94.67**	**86.46**	**85.19**	**92.59**
Q+V+Conv(N=2,2,2)	1.728	4.664	89.22	54.75	70.86	77.99	94.61	86.30	85.00	92.50
Q+V+Conv(N=4,2,1)	**1.705**	**4.618**	**89.22**	**56.67**	**72.23**	**79.27**	94.60	86.22	84.90	92.45

potential correlation between HE and SS, we introduce the FEA-Gate, which incorporates long-range dependencies of SS into HE, effectively regularizing the HE process. Additionally, height maps are processed through a prompt encoder to obtain height vectors, which guide the generation of the segmentation map in the SS branch. Experiments on multiple datasets demonstrate that HEASSNet achieves superior performance and compatibility. In future work, We will further explore dense and small object recognition to improve the performance of SS and HE tasks.

REFERENCES

[1] P. Ghamisi and N. Yokoya, "Img2dsm: Height simulation from single imagery using conditional generative adversarial net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 794–798, 2018.

[2] M. E. Paoletti, J. M. Haut, P. Ghamisi, N. Yokoya, J. Plaza, and A. Plaza, "U-img2dsm: Unpaired simulation of digital surface models with generative adversarial networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 7, pp. 1288–1292, 2020.

[3] X. Li, G. Zhang, H. Cui, S. Hou, S. Wang, X. Li, Y. Chen, Z. Li, and L. Zhang, "Mcanet: A joint semantic segmentation framework of optical and sar images for land use classification," *International Journal of Applied Earth Observation and Geoinformation*, vol. 106, p. 102638, 2022.

[4] D. Hong, B. Zhang, H. Li, Y. Li, J. Yao, C. Li, M. Werner, J. Chanussot, A. Zipf, and X. X. Zhu, "Cross-city matters: A multimodal remote sensing benchmark dataset for cross-city semantic segmentation using high-resolution domain adaptation networks," *Remote Sensing of Environment*, vol. 299, p. 113856, 2023.

[5] M. Li, J. Long, A. Stein, and X. Wang, "Using a semantic edge-aware multi-task neural network to delineate agricultural parcels from remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 200, pp. 24–40, 2023.

[6] R. Li, D. Xue, S. Su, X. He, Q. Mao, Y. Zhu, J. Sun, and Y. Zhang, "Learning depth via leveraging semantics: Self-supervised monocular depth estimation with both implicit and explicit semantic guidance," *Pattern Recognition*, vol. 137, p. 109297, 2023.

[7] F. Li, H. Zhang, H. Xu, S. Liu, L. Zhang, L. M. Ni, and H.-Y. Shum, "Mask dino: Towards a unified transformer-based framework for object detection and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3041–3050.

[8] Y. Chen, D. Zhao, L. Lv, and Q. Zhang, "Multi-task learning for dangerous object detection in autonomous driving," *Information Sciences*, vol. 432, pp. 559–571, 2018.

[9] C. Chen, C. Wang, B. Liu, C. He, L. Cong, and S. Wan, "Edge intelligence empowered vehicle detection and image segmentation for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 11, pp. 13 023–13 034, 2023.

[10] H. Guo, Q. Shi, B. Du, L. Zhang, D. Wang, and H. Ding, "Scene-driven multitask parallel attention network for building extraction in high-resolution remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4287–4306, 2020.

[11] J. Feng, Q. Jiang, C.-H. Tseng, X. Jin, L. Liu, W. Zhou, and S. Yao, "A deep multitask convolutional neural network for remote sensing image super-resolution and colorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[12] Y. Sun, X. Zhang, J. Huang, H. Wang, and Q. Xin, "Fine-grained building change detection from very high-spatial-resolution remote sensing images based on deep multitask learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2020.

[13] J. Zhu, Y. Zhou, N. Xu, and C. Huo, "Collaborative learning network for change detection and semantic segmentation of remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, 2023.

[14] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[15] C. P. R. H. G. V. K. C. Z. Y. D. T. H. H. M. S. X. Sun, "2023 ieee grss data fusion contest: Large-scale fine-grained building classification for semantic urban reconstruction," 2022. [Online]. Available: https://dx.doi.org/10.21227/mrnt-8w27

[16] K. Yang, G.-S. Xia, Z. Liu, B. Du, W. Yang, M. Pelillo, and L. Zhang, "Semantic change detection with asymmetric siamese networks," 2020.

[17] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[18] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7151–7160.

[19] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[20] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3146–3154.

[21] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," in *European conference on computer vision*. Springer, 2022, pp. 205–218.

[22] L. Wang, R. Li, C. Zhang, S. Fang, C. Duan, X. Meng, and P. M. Atkinson, "Unetformer: A unet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 190, pp. 196–214, 2022.

[23] L. Mou and X. X. Zhu, "Im2height: Height estimation from single monocular imagery via fully residual convolutional-deconvolutional network," *arXiv preprint arXiv:1802.10249*, 2018.

[24] S. Du, J. Xing, S. Wang, X. Xiao, J. Li, and H. Liu, "Lumnet: Land use knowledge guided multiscale network for height estimation from single remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, 2024.

[25] S. Chen, Y. Shi, Z. Xiong, and X. X. Zhu, "Htc-dc net: Monocular height estimation from single remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–18, 2023.

[26] A. Agarwal and C. Arora, "Attention attention everywhere: Monocular depth prediction with skip attention," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 5861–5870.

[27] D. Kim, W. Ga, P. Ahn, D. Joo, S. Chun, and J. Kim, “Global-local path networks for monocular depth estimation with vertical cutdepth. arxiv 2022,” *arXiv preprint arXiv:2201.07436*.

[28] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, “Neural window fully-connected crfs for monocular depth estimation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3916–3925.

[29] L. Ding, J. Zhang, H. Guo, K. Zhang, B. Liu, and L. Bruzzone, “Joint spatio-temporal modeling for semantic change detection in remote sensing images,” *IEEE Transactions on Geoscience and Remote Sensing*, 2024.