HSGraphAgent: Knowledge-Graph-Guided Large Language Models for Harmonized System Code Classification

Qiang Xia[1,2†], **Zijian Zhang**[3†], **Ao Wang**[2], **Wenhan Wang**[3], **Xiangyu Wang**[3], **Jian Li**[1,2*]
[1]Urban Mobility Institute, Tongji University
[2]The Key Laboratory of Road and Traffic Engineering, Ministry of Education
[3]COSCO SHIPPING Technology Co., Ltd.
xiaqiang@tongji.edu.cn; zhang.zijian@coscoshipping.com; 2331623@tongji.edu.cn;
wang.wenhan@coscoshipping.com; wang.xiangyu@coscoshipping.com; jianli@tongji.edu.cn

Abstract

Harmonized System (HS) code classification is a hierarchically structured and regulation-constrained task, often complicated by short and noisy product descriptions. Misclassification can lead to tariff misapplication, regulatory violations, or delayed customs clearance; predictions therefore need to be both semantically appropriate and hierarchically valid. While large language models (LLMs) show strong semantic understanding, their unconstrained generation is poorly aligned with these requirements, often producing non-existent or hierarchically inconsistent codes. We propose **HSGraphAgent**, a knowledge-graph-guided LLM framework that formulates HS classification as a stepwise, regulation-aware reasoning process over an explicit HS knowledge graph. By encoding hierarchical containment relations and regulatory exclusion rules, and enforcing them through a *Select–Redirect* mechanism, HSGraphAgent constrains inference to legally valid paths while producing explicit and traceable reasoning trajectories. Experiments on taxonomy-wide 4-digit and fine-grained 6-digit HS benchmarks demonstrate consistent improvements over direct generation and retrieval-augmented baselines, with particularly strong gains in fine-grained and regulation-sensitive classification settings.

1 Introduction

In contemporary trade systems, virtually all import and export transactions require products to be assigned Harmonized System (HS) codes as part of mandatory customs declarations. Given the sheer scale of national trade volumes, HS code assignment constitutes a high-frequency and large-scale decision process that directly affects customs clearance, tariff application, regulatory compliance, and trade statistics (World Customs Organization, 2022).

Despite the availability of large-scale historical declaration data, HS classification in practice still relies heavily on manual expertise (Chen et al., 2021). The core limitation is that critical regulatory knowledge is rarely encoded in machine-actionable form. This includes hierarchical containment logic and exclusion rules defined in tariff notes. In real-world workflows, experts classify products by traversing the HS hierarchy from top to bottom. At each level, candidate categories are validated against regulatory conditions, and decisions are revised when exclusions apply. This reasoning process is difficult to automate, especially when product descriptions are short, noisy, or non-standardized.

Most automated approaches, including traditional machine learning and neural classifiers, treat HS codes as flat or weakly structured labels at inference time (Ding et al., 2015; Altaheri and Shaalan, 2020). Retrieval-based and hierarchical models improve semantic alignment but still lack mechanisms to guarantee global path consistency or enforce regulatory exclusions during inference (Anggoro et al., 2025).

Recent advances in large language models (LLMs) offer strong semantic understanding of unstructured product descriptions (Achiam et al., 2023; Chang et al., 2024), making them appealing for HS classification. However, standard LLM inference relies on unconstrained generation over an open output space, which does not naturally match the rigid, regulation-constrained structure of the HS system. In practice, LLMs may hallucinate non-existent codes, skip hierarchy levels, or ignore exclusion rules that are critical for legal compliance.

A common strategy to mitigate these issues is retrieval-augmented generation (RAG), which supplies LLMs with relevant external documents at inference time (Lewis et al., 2020; Fan et al., 2024). While retrieval can improve semantic grounding

[†]These authors contributed equally.
[*]Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 44761–44773
July 2-7, 2026 ©2026 Association for Computational Linguistics

and reduce hallucination, it does not impose explicit constraints on the reasoning process. As a result, RAG-based approaches still operate over locally retrieved candidates and cannot guarantee global path consistency or enforce regulatory exclusions during hierarchical decision making (Wang et al., 2025). This limitation becomes particularly pronounced in fine-grained classification, where legally adjacent categories exhibit high semantic similarity but differ in regulatory scope.

Progress toward structured and regulation-aware HS classification is further limited by gaps in existing resources and evaluation. To the best of our knowledge, publicly available HS resources rarely provide a machine-readable representation that jointly encodes hierarchical structure and explicit regulatory exclusion logic. As a result, most existing approaches rely on implicit knowledge embedded in model parameters or unstructured text, limiting transparency and controllability. Moreover, commonly used datasets often cover only limited portions of the HS taxonomy or exhibit skewed distributions across chapters (Du et al., 2021; Lee et al., 2024), making them insufficient for evaluating hierarchical reasoning and fine-grained classification performance.

To address these limitations, we propose **HS-GraphAgent**[1], a knowledge-graph-guided LLM framework for hierarchical HS code classification. Rather than directly generating HS codes, HS-GraphAgent formulates the task as a constrained graph traversal problem over an explicit HS knowledge graph that encodes legal containment relations and exclusion rules. An LLM performs stepwise, top-down classification over the HS hierarchy, with constraint checks applied at each level. The *Select–Redirect* mechanism guides candidate selection and triggers redirection when regulatory constraints are violated, thereby enforcing hierarchical consistency and reducing invalid predictions.

Our contributions are three-fold:

• **HS Knowledge Graph with Explicit Constraints.** We construct a machine-readable HS knowledge graph that encodes hierarchical containment together with regulation-driven exclusion and redirection rules, enabling auditable and constraint-aware classification.

• **Benchmarks for Hierarchical and Constraint-Aware Reasoning.** We introduce a taxonomy-wide 4-digit benchmark and a realistic 6-digit benchmark to evaluate hierarchical consistency and reasoning under structural and regulatory constraints.

• **Graph-Guided LLM Inference via Select–Redirect Reasoning.** We propose a knowledge-graph-guided LLM inference framework that enforces hierarchical validity and regulatory compliance through a stepwise Select–Redirect mechanism, producing valid and traceable decision paths.

The framework shows how explicit legal constraints can be enforced during LLM inference for hierarchical classification tasks.

2 Related Work

Automatic HS code classification has been studied extensively in intelligent trade and customs systems. The task is characterized by short and noisy product descriptions and a legally defined label space with strict hierarchical and regulatory constraints.

Recent work has explored various mechanisms for constraining the outputs of LLMs. These include constrained decoding (Geng et al., 2023), schema-guided generation (Zhang et al., 2025), and tool-augmented inference frameworks (Zhuang et al., 2023). In parallel, knowledge graphs have been integrated with LLMs to support structured reasoning beyond unstructured text, such as graph-augmented prompting and neuro-symbolic inference (Sun et al., 2024; Jiang et al., 2025). While these approaches improve controllability or incorporate symbolic structure, they primarily focus on output formatting, retrieval augmentation, or representation fusion. Explicit modeling of constraint-aware traversal over hierarchical decision spaces during inference remains limited.

HS code classification has traditionally been approached as a large-scale text classification or decision support problem. Early studies relied on conventional supervised models trained on historical customs data (Singh and Sahu, 2004; Mukherjee et al., 2008). Subsequent work introduced retrieval-based methods and hierarchical modeling techniques to improve semantic matching and consistency across classification levels (Ding et al., 2015; Spichakova and Haav, 2020; He et al., 2021). More recent studies have explored structured knowledge integration and LLM-based generation or reranking strategies for HS classification (Sun et al., 2025; Navasardyan, 2024; Koch and Power, 2025). De-

[1]Dataset page: https://github.com/VoldeMorddddddd/HSBench.

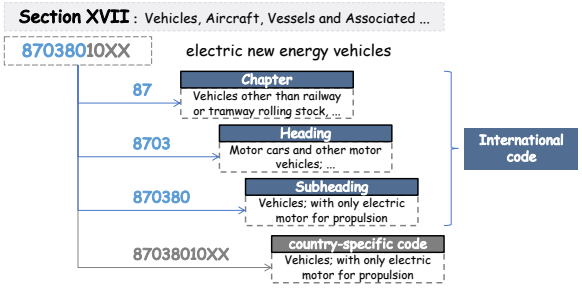

Figure 1: Hierarchical structure of the international HS code system, illustrating prefix-based top-down classification under strict hierarchical constraints.

spite these advances, most existing methods lack explicit and auditable inference processes that enforce hierarchical path consistency and regulatory exclusion logic, particularly in fine-grained classification settings (Lee et al., 2024).

Prior work has improved semantic modeling for HS classification, but inference-time enforcement of hierarchical and regulatory constraints remains underexplored, especially in settings where invalid classification paths incur significant operational risk.

3 Methodology

We propose HSGraphAgent, a knowledge-graph-guided LLM framework that formulates HS code classification as a hierarchically constrained graph reasoning problem. The framework integrates semantic inference with explicit hierarchical and regulatory constraints to enable stable and interpretable classification.

3.1 Preliminary

Harmonized System. The Harmonized System (HS) is an internationally standardized nomenclature for classifying traded products, developed and maintained by the World Customs Organization (WCO) and adopted by over 200 countries and economies. HS codes are organized as a strictly hierarchical taxonomy, as shown in Figure 1.

Task definition. We model the HS system as a directed graph

$$G = (V, E_c, E_r), \tag{1}$$

where V denotes the set of nodes corresponding to valid HS entries. The edge set E_c represents containment relations that define the hierarchical structure of the HS taxonomy. The edge set E_r represents regulatory relations derived from tariff notes, encoding condition-triggered exclusion or redirection constraints. A regulatory relation is defined as a triple (v_i, ϕ, v_j), which specifies that products satisfying condition ϕ must not be classified under node v_i and should instead be redirected to node v_j.

Given a product description x, the task is to identify a terminal node

$$v^* \in V_d, \quad d \in \{4, 6\}, \tag{2}$$

corresponding to the target classification depth. The selected node must be reachable from the root via containment relations, satisfy all applicable regulatory constraints, and be semantically consistent with x. Equivalently, HS classification is formulated as the problem of finding a top-down path in G that is both semantically valid and compliant with hierarchical and regulatory constraints.

3.2 Overview

Figure 2 provides an overview of HSGraphAgent, which formulates HS code classification as a hierarchical, regulation-aware reasoning process over an explicit knowledge graph. Given a product description, the framework predicts an HS code together with a traceable reasoning path, rather than producing a single-shot label prediction.

HSGraphAgent consists of two tightly coupled components. First, we construct an explicit HS knowledge graph that encodes both the legally defined hierarchical structure of the HS system and regulatory exclusion or redirection rules derived from tariff notes. This graph defines the valid decision space for classification and serves as the structural backbone for inference.

Second, classification is performed via graph-guided reasoning over the constructed knowledge graph. An LLM agent traverses the HS hierarchy in a top-down manner, selecting candidate nodes under hierarchical constraints and validating each decision against regulatory rules. A *Select–Redirect* loop enforces constraint-aware traversal by correcting invalid decisions at inference time and guiding the reasoning process toward a compliant terminal node at the target depth.

3.3 HS Knowledge Graph Construction

We construct an HS knowledge graph that combines the HS taxonomy with regulatory exclusion and redirection rules, providing the constrained decision space used by HSGraphAgent during inference.

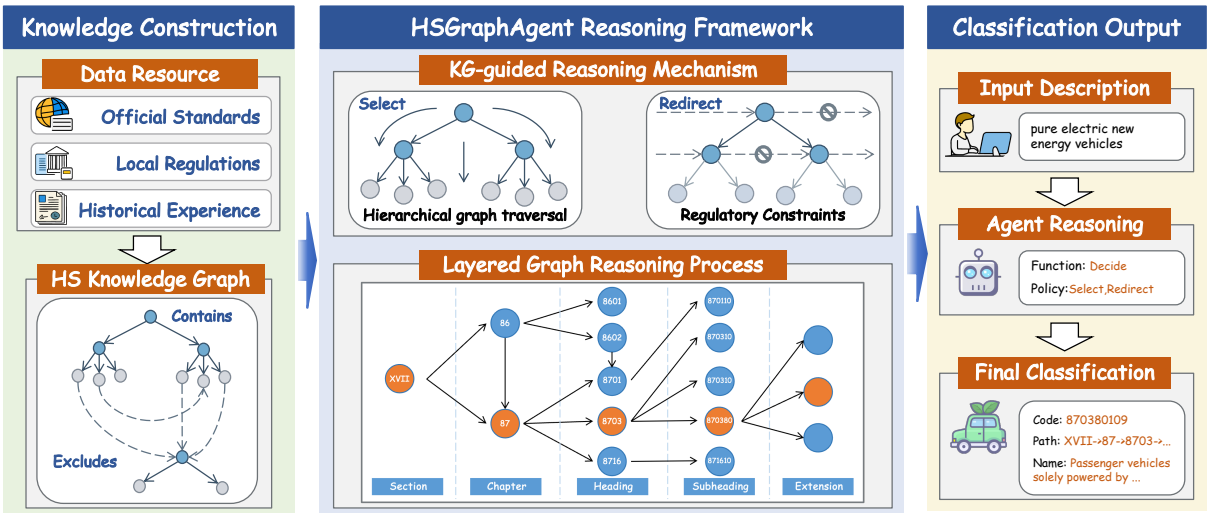

Figure 2: HSGraphAgent framework for hierarchical and regulation-aware HS code classification.

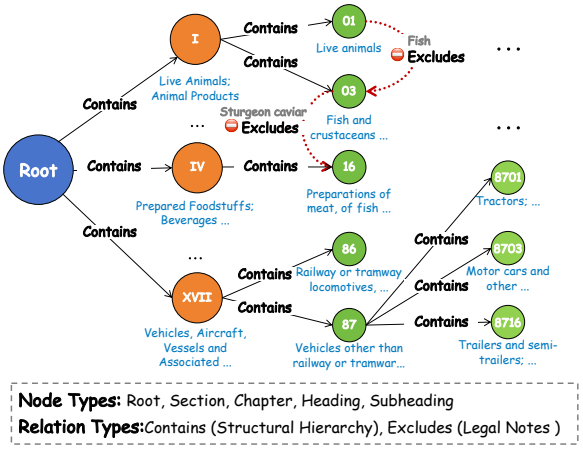

Figure 3: Illustration of the constructed HS knowledge graph.

Data sources. The HS knowledge graph is constructed from authoritative regulatory sources. We use official documentation released by the WCO to obtain the globally harmonized HS hierarchy and associated legal notes. To capture fine-grained exclusion rules, we additionally incorporate the China Import and Export Tariff Schedule, which provides detailed explanatory texts at the subheading level. Although tariff schedules may include jurisdiction-specific extensions, the graph construction process is modular and allows such rules to be incorporated without modifying the underlying hierarchical structure.

Extraction pipeline. Regulatory knowledge is not manually authored rule by rule. Instead, we use an LLM-assisted extraction pipeline to convert official tariff texts into machine-readable graph elements. For each note or explanatory passage, the pipeline identifies the relevant HS node, extracts inclusion or exclusion conditions, maps redirected targets when applicable, and normalizes the result into containment or regulatory edges. The extracted structures are then checked against the official HS hierarchy to ensure that node identifiers and redirection targets are legally valid before being added to the graph. This design reduces manual authoring effort and makes the graph construction process easier to reproduce and extend to new tariff documents.

Hierarchical modeling. We encode the HS taxonomy as a directed, multi-level graph that follows the legally defined top-down hierarchy from coarse to fine categories. Each node represents a valid HS entry and is associated with its code, official name, and descriptive text extracted from tariff notes. Directed *containment* edges capture parent–child relationships across levels, forming a tree-structured backbone rooted at the global HS root. This hierarchical backbone constrains all reasoning trajectories to legally valid classification paths and prevents invalid jumps across levels during inference.

Regulatory exclusion and redirection modeling. Beyond hierarchical containment, the HS system specifies exclusion and redirection rules through tariff notes, which we encode as directed regulatory edges in the knowledge graph. These edges represent condition-triggered constraints indicating that products satisfying certain criteria must not be classified under a given node, but should instead be redirected to a legally valid alternative.

Table 1: Structural statistics of the constructed HS knowledge graph. "Contains" and "Excludes" denote the average numbers of containment and exclusion edges per node, respectively.

Node Type	#Nodes	Contains	Excludes
Section	22	4.41	3.14
Chapter	97	12.67	5.53
Heading	1231	4.56	4.47
Subheading	5615	2.13	0.00

As illustrated in Figure 3, although Chapter 03 (Fish and crustaceans) is hierarchically compatible with products described as "fish," tariff notes exclude processed fish products and redirect them to Chapter 16 (Preparations of meat or fish). Such constraints cannot be inferred from semantic similarity or hierarchical proximity alone. Explicitly modeling regulatory exclusion and redirection is therefore essential for preventing locally plausible but legally invalid classification paths during inference.

Graph summarization. To support consistent step-wise reasoning, we attach concise semantic summaries to graph nodes. These summaries encode inclusion scope and exclusion criteria derived from tariff notes, providing a compact representation of containment semantics. Summaries are generated offline and used as auxiliary context during inference, enabling the model to evaluate candidate nodes under hierarchical and regulatory constraints without accessing raw tariff texts.

Table 1 summarizes the topology of the constructed HS knowledge graph across hierarchical levels. Chapters have the highest average containment and exclusion connectivity, reflecting their role in separating broad product categories before downstream classification. Headings concentrate much of the exclusion logic, with nearly as many exclusion edges as containment edges on average, indicating that medium-grained decisions are where semantic similarity alone is most likely to conflict with regulatory scope. By contrast, subheadings behave as terminal leaves with no exclusion edges and a low average degree, which is consistent with their role as fine-grained endpoints in official HS documentation. Overall, this distribution shows that exclusion rules are primarily defined at higher levels of the hierarchy, while lower levels mainly refine legally valid paths that have already been established upstream.

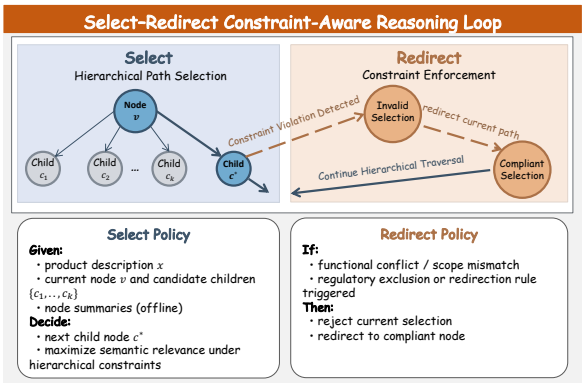

Figure 4: The Select–Redirect reasoning loop of HSGraphAgent.

3.4 Graph-Guided Constraint-Aware Reasoning

HSGraphAgent formulates HS code classification as a constrained sequential decision process over an explicit HS knowledge graph. Rather than directly generating a terminal HS code, the agent incrementally constructs a classification path by traversing the HS hierarchy from the root to the target depth. At each step, decisions are jointly governed by semantic relevance, hierarchical validity, and explicit regulatory constraints, ensuring that inference remains aligned with the legal structure of the HS system.

Formally, given a product description x and an HS knowledge graph $G = (V, E_c, E_r)$, inference proceeds as a path construction process

$$\pi = (v_0, v_1, \ldots, v_t), \tag{3}$$

where v_0 denotes the root node of the HS hierarchy and each transition $(v_i, v_{i+1}) \in E_c$ follows a legally defined containment relation. The objective is to identify a terminal node v_t at the desired classification depth such that the resulting path π is semantically consistent with the input description x and does not violate any regulatory constraints encoded in E_r. This formulation explicitly models HS classification as a hierarchical reasoning problem with inference-time constraint enforcement.

To operationalize this process, HSGraphAgent employs an iterative *Select–Redirect* reasoning loop that tightly couples LLM inference with structured constraint checking. Unlike flat prediction or unconstrained decoding, this loop enforces hierarchical and regulatory validity *during* the reasoning process itself, rather than relying on post hoc validation.

Select policy (hierarchically constrained selection). At a current node v_i along the partial path $\pi_{0:i}$, the agent considers the set of candidate child nodes

$$\mathcal{C}(v_i) = \{c \mid (v_i, c) \in E_c\}. \tag{4}$$

The LLM is prompted with (i) the product description x, (ii) the current reasoning path $\pi_{0:i}$, and (iii) concise semantic summaries of nodes in $\mathcal{C}(v_i)$. Conditioned on this information, the model selects the next node

$$v_{i+1} = \arg\max_{c \in \mathcal{C}(v_i)} p(c \mid x, \pi_{0:i}), \tag{5}$$

where the probability is implicitly represented by the model's preference under the constrained candidate set. Crucially, the selection space is restricted to locally valid child nodes defined by the HS hierarchy, preventing invalid jumps across levels and ensuring that all intermediate decisions respect the legally defined top-down structure. When no candidate child yields a sufficiently plausible continuation, the agent may return to the previous valid node and resume traversal from a higher level. This fallback is treated as a controlled backtracking operation rather than unconstrained path revision.

Redirect policy (regulatory constraint enforcement). After a candidate node v_{i+1} is selected, HSGraphAgent evaluates whether the extended path $\pi_{0:i+1}$ violates any regulatory constraints encoded in E_r. Each regulatory edge $(v, \phi, u) \in E_r$ specifies a condition ϕ under which classification under node v is legally invalid and must be redirected to node u. If the current selection triggers such a violation, the agent rejects v_{i+1} and performs a redirection by updating the reasoning path to a compliant alternative.

Redirection operates at the *path level* rather than the node level. This allows the agent to recover from locally plausible but globally invalid decisions, preventing irreversible error propagation that commonly arises in greedy or unconstrained stepwise traversal. By explicitly encoding regulatory logic in the graph and enforcing it during inference, HSGraphAgent ensures that all accepted paths remain legally valid. If neither direct continuation nor redirection yields a compliant next step, the agent falls back to the previous valid node and continues reasoning from that point.

Iterative reasoning and termination. The Select–Redirect loop is repeated as the agent traverses the HS hierarchy from coarse-grained to fine-grained levels. Reasoning terminates when either (i) the target classification depth is reached, or (ii) no further compliant expansion is possible under the regulatory constraints. The final output consists of the terminal HS code together with the complete reasoning path π, providing an explicit and auditable account of the classification decision.

Algorithm 1 Select–Redirect Reasoning in HSGraphAgent

Require: Product description x, HS graph $G = (V, E_c, E_r)$, target depth d

Initialize $v \leftarrow v_0$ (root), path $\pi \leftarrow [v]$
while depth$(v) < d$ **do**
  $\mathcal{C} \leftarrow \{c \mid (v, c) \in E_c\}$
  $c^* \leftarrow$ LLM_Select$(x, \pi, \mathcal{C})$
  if $c^* =$ BACKTRACK **then**
    $v \leftarrow$ Parent(v)
  else if violates_regulation$(\pi \cup \{c^*\}, E_r)$ **then**
    $v \leftarrow$ RedirectOrBacktrack(π, c^*, E_r)
  else
    $v \leftarrow c^*$
  Append v to π
return terminal node v and reasoning path π

4 Experiments

We evaluate HSGraphAgent on hierarchical HS code classification at different levels of granularity under a zero-shot setting. The experiments compare HSGraphAgent with representative LLM and retrieval-augmented baselines, focusing on how performance changes with classification depth and inference-time constraint enforcement.

4.1 Experimental Setup

4.1.1 Datasets

We construct two datasets for hierarchical HS code classification at different levels of granularity. Both datasets are derived from real-world product descriptions and manually verified to ensure consistency with official HS definitions.

1) 4-digit HS Code Full-Coverage Dataset (1,231 instances). This dataset provides full coverage of all 4-digit HS codes at the heading level. Product descriptions are collected from user queries and general product descriptions and are typically short and loosely structured. This setting reflects a common medium-grained classification scenario, where inputs contain limited technical

detail and the goal is to identify the correct tariff heading.

2) 6-digit HS Code Test Dataset (450 instances). This dataset consists of a representative subset of 6-digit HS codes. Product descriptions are closer to actual customs declarations and typically include discriminative attributes such as usage, material composition, and processing characteristics. This dataset evaluates fine-grained classification under stricter hierarchical and regulatory constraints.

4.1.2 Compared Methods

We compare HSGraphAgent with the following baselines.

• **Only LLM.** The language model directly generates HS codes from product descriptions without access to explicit hierarchical structure or regulatory constraints.

• **RAG-Name.** A retrieval-augmented baseline that retrieves HS entries using vector embeddings constructed from code–name pairs.

• **RAG-Window.** A retrieval-augmented baseline that extends name embeddings with windowed descriptive texts. Text is segmented using a sliding window of 400 tokens with a stride of 50.

• **HSGraphAgent (Ours).** A graph-guided hierarchical reasoning framework that performs constrained top-down traversal over an explicit HS knowledge graph. The framework enforces hierarchical and regulatory constraints during inference using a Select–Redirect reasoning loop.

All methods operate under a zero-shot setting and use comparable textual information. They differ only in whether hierarchical and regulatory constraints are explicitly enforced during inference. We focus on the zero-shot setting to isolate the effect of inference-time structural constraint enforcement.

4.1.3 Evaluation Metrics and Protocols

All models are evaluated in the same zero-shot setting. For retrieval-based methods, we use the `bge-base-zh-v1.5` embedding model and retrieve the top-10 most relevant text chunks per query. Performance is measured using hierarchical Top-1 accuracy at the HS-2, HS-4, and HS-6 levels. A prediction is considered correct at a given level if the corresponding HS prefix matches the ground truth.

Qwen2.5-32B is deployed locally on a server equipped with two NVIDIA A100-PCIE GPUs, each with 40GB of memory. The system uses 20 vCPUs based on Intel Xeon (Skylake, IBRS) processors and 144GB of system memory. All inference experiments are conducted under identical hardware conditions.

4.2 Performance Comparison

We evaluate HSGraphAgent across four representative LLM backbones with varying reasoning capacities: DeepSeek-V3.2-685B (Liu et al., 2025), Kimi-K2-1T (Team et al., 2025), GPT-OSS-120B (Agarwal et al., 2025), and Qwen2.5-32B (Qwen et al., 2025).

Table 2 reports hierarchical Top-1 accuracy at the HS-2, HS-4, and HS-6 levels on both the 4-digit and 6-digit datasets. Several consistent patterns emerge across models and inference strategies.

Retrieval-based methods work well at medium granularity but saturate at finer levels. On the 4-digit dataset, both RAG-Name and RAG-Window consistently outperform direct generation across all backbones. Medium-grained heading identification can often be resolved through local semantic similarity. Despite higher accuracy at the HS-2 and HS-4 levels, retrieval-based methods show limited gains at HS-6. This suggests that fine-grained classification increasingly depends on global hierarchical consistency, rather than local semantic matching alone.

The effectiveness of retrieval is strongly modulated by the reasoning capacity of the backbone LLM. For models with weaker intrinsic reasoning abilities, such as GPT-OSS-120B and Qwen2.5-32B, retrieval yields large improvements over direct generation but plateaus once relevant candidates are retrieved. In contrast, stronger backbones such as Kimi-K2-1T and DeepSeek-V3.2-685B are better able to exploit retrieved context, achieving higher overall accuracy. Nevertheless, even with these models, retrieval alone remains insufficient to ensure consistent improvements at the HS-6 level, highlighting the limitation of retrieval-based approaches in enforcing global hierarchical validity.

HSGraphAgent consistently achieves the strongest fine-grained performance by enforcing hierarchical and regulatory constraints during inference. Across all backbone models, HSGraphAgent attains the highest HS-6 accuracy, with substantial absolute improvements over the strongest retrieval-based baseline. Specifically, HSGraphAgent improves HS-6 accuracy by 18.0 points on DeepSeek-V3.2-685B (0.902 vs. 0.722),

Table 2: Hierarchical Top-1 accuracy on the 4-digit and 6-digit HS code datasets across different backbone LLMs. Best results within each model block are highlighted in bold.

Model	Method	4-digit Dataset		6-digit Dataset		
		HS-2	HS-4	HS-2	HS-4	HS-6
DeepSeek-V3.2-685B	Only LLM	0.928	0.791	0.934	0.802	0.489
	RAG-Name	0.936	0.891	0.902	0.827	0.702
	RAG-Window	0.958	0.918	0.907	0.833	0.722
	HSGraphAgent	**0.973**	**0.969**	**0.965**	**0.940**	**0.902**
Kimi-K2-1T	Only LLM	0.936	0.819	0.938	0.844	0.553
	RAG-Name	0.957	0.890	0.927	0.867	0.758
	RAG-Window	**0.958**	0.908	0.938	0.896	0.789
	HSGraphAgent	0.949	**0.920**	**0.962**	**0.942**	**0.909**
GPT-OSS-120B	Only LLM	0.761	0.198	0.722	0.309	0.076
	RAG-Name	0.911	0.860	0.851	0.782	0.669
	RAG-Window	**0.938**	**0.885**	0.900	0.804	0.696
	HSGraphAgent	0.901	0.818	**0.911**	**0.862**	**0.789**
Qwen2.5-32B	Only LLM	0.726	0.200	0.645	0.266	0.038
	RAG-Name	0.927	0.860	0.842	0.784	0.658
	RAG-Window	**0.940**	**0.882**	0.864	0.778	0.676
	HSGraphAgent	0.894	0.859	**0.896**	**0.860**	**0.778**

12.0 points on Kimi-K2-1T (0.909 vs. 0.789), 9.3 points on GPT-OSS-120B (0.789 vs. 0.696), and 10.2 points on Qwen2.5-32B (0.778 vs. 0.676). These gains persist even for strong backbone models, indicating that the improvements stem from inference-time enforcement of hierarchical and regulatory constraints rather than increased model capacity alone.

Overall, these results demonstrate that while retrieval provides strong local semantic signals, it cannot guarantee global structural coherence. Explicit integration of hierarchical structure and regulatory constraints into the inference process is essential for robust and legally compliant HS code classification, especially in fine-grained settings.

4.3 Efficiency Analysis

Table 3 further compares predictive performance and inference cost under a shared DeepSeek-V3.2-685B backbone. The results indicate that inference cost should be considered together with the reliability required by the task. Retrieval-based methods improve accuracy over direct generation with only modest increases in latency and token consumption, making them attractive when response cost is the primary concern. Compared with the best retrieval baseline, HSGraphAgent improves 6-digit accuracy from 0.72 to 0.90 while increasing latency from 8.68 s to 24.97 s. In HS code classification, such gains are important because an incorrect fine-grained code can affect tariff treatment, compliance review, and downstream regulatory decisions. This pattern indicates that the additional inference cost is most justified when reliable fine-grained classification is required.

4.4 Ablation Study

We conduct ablation experiments using DeepSeek-V3.2-685B as the backbone LLM to assess the contribution of key components in HSGraphAgent. Specifically, we isolate the effects of (i) global structural awareness and (ii) regulatory constraint enforcement within the Select–Redirect reasoning framework.

We evaluate two ablated variants: **(1) w/o Redirect**, which disables regulatory redirection and allows unconstrained step-wise traversal; and **(2) w/o Abstract**, which removes global graph summaries and restricts reasoning to local node-level information.

Table 4 reports the ablation results. Removing either component leads to a clear performance degradation, with the most pronounced effects observed at finer-grained levels.

Impact of regulatory redirection. Disabling the Redirect mechanism results in a substantial drop in HS-6 accuracy. Once an invalid hierarchical path is selected, unconstrained step-wise reasoning

Table 3: Accuracy–efficiency comparison on DeepSeek-V3.2-685B. Accuracy is reported at HS-4 for the 4-digit dataset and HS-6 for the 6-digit dataset.

Method	4-digit Dataset			6-digit Dataset		
	Acc	Latency (s)	Tokens	Acc	Latency (s)	Tokens
Only LLM	0.79	7.27	403	0.49	10.49	584
RAG-Name	0.89	8.34	631	0.70	9.10	682
RAG-Window	0.92	8.15	1195	0.72	8.68	1041
HSGraphAgent	0.97	19.25	8277	0.90	24.97	7702

Table 4: Ablation results of HSGraphAgent on hierarchical HS code classification.

Method	4-digit Dataset		6-digit Dataset		
	HS-2	HS-4	HS-2	HS-4	HS-6
w/o Redirect	0.839	0.797	0.776	0.729	0.669
w/o Abstract	0.858	0.797	0.831	0.747	0.698
Full	**0.973**	**0.969**	**0.965**	**0.940**	**0.902**

cannot recover, leading to irreversible downstream errors. This confirms that corrective redirection is essential for enforcing regulatory compliance and maintaining path validity during inference.

Impact of global graph summarization. Removing global summaries primarily affects HS-4 and HS-6 performance. Without a high-level structural overview, the model struggles to maintain awareness of its position within the HS hierarchy, resulting in suboptimal candidate selection even when local semantic cues are available.

5 Conclusion

We presented HSGraphAgent, a knowledge-graph-guided LLM framework for hierarchical HS code classification under strict regulatory constraints. By formulating classification as a stepwise, constraint-aware traversal over an explicit HS knowledge graph, the proposed approach aligns LLM inference with the legal structure of the HS system rather than relying on post hoc validation or retrieval alone. Empirical results on 4-digit and 6-digit benchmarks show that enforcing hierarchical and regulatory constraints during inference yields more stable and accurate predictions, especially in fine-grained and regulation-sensitive settings where retrieval-based methods show limited fine-grained gains.

The same design may also apply to other hierarchical classification tasks where legal or procedural rules must be enforced during inference.

6 Limitations

HSGraphAgent is designed for classification systems governed by explicit hierarchical taxonomies and regulation-based exclusion rules. While the Harmonized System provides a globally standardized structure up to the 6-digit level, jurisdiction-specific extensions and interpretive notes beyond that level may vary across customs systems. In our current instantiation, the hierarchical backbone follows the globally shared WCO standard, while fine-grained regulatory annotations are derived from the China Import and Export Tariff Schedule. As a result, transferring the framework to other national systems such as US HTS or EU CN would mainly require replacing or augmenting the local regulatory layer, rather than redesigning the underlying reasoning framework.

In addition, regulatory relations are derived from natural-language tariff documents. The completeness and specificity of such rules are uneven across HS categories, which may affect the degree of constraint enforcement achievable for certain products. This limitation reflects the variability of regulatory documentation rather than a property of the reasoning framework itself.

Finally, graph-guided reasoning involves multi-step traversal over the HS hierarchy, which introduces additional inference cost compared with single-pass generation or retrieval-based approaches. This trade-off is inherent to enforcing hierarchical and regulatory consistency during inference. Future work may explore efficiency optimizations and adaptive reasoning strategies to support large-scale or time-sensitive deployment scenarios.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman,

Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Fatma Altaheri and Khaled Shaalan. 2020. Exploring machine learning models to predict harmonized system code. In *Information Systems*, volume 381, pages 291–303. Springer International Publishing.

Angga Wahyu Anggoro, Padraig Corcoran, Dennis De Widt, and Yuhua Li. 2025. Harmonized system code classification using supervised contrastive learning with sentence BERT and multiple negative ranking loss. *Data Technologies and Applications*, 59(2):276–301.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

Xi Chen, Stefano Bromuri, and Marko Van Eekelen. 2021. Neural machine translation for harmonized system codes prediction. In *2021 6th International Conference on Machine Learning Technologies*, pages 158–163. ACM.

Liya Ding, ZhenZhen Fan, and DongLiang Chen. 2015. Auto-categorization of hs code using background net approach. *Procedia Computer Science*, 60:1462–1471.

Shaohua Du, Zhihao Wu, Huaiyu Wan, and YouFang Lin. 2021. HScodeNet: Combining hierarchical sequential and global spatial information of text for commodity HS code classification. In *Advances in Knowledge Discovery and Data Mining*, volume 12713, pages 676–689. Springer International Publishing.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6491–6501.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured nlp tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10932–10952. Association for Computational Linguistics.

Mingshu He, Xiaojuan Wang, Chundong Zou, Bingying Dai, and Lei Jin. 2021. A commodity classification framework based on machine learning for analysis of trade declaration. *Symmetry*, 13(6):964.

Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, Yang Song, Chen Zhu, Hengshu Zhu, and Ji-Rong Wen. 2025. Kg-agent: An efficient autonomous agent framework for complex reasoning over knowledge graph. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9505–9523.

Thomas Koch and Kevin Power. 2025. Automating harmonized system (HS) code classification from unstructured shipping manifests using large language models. In *2025 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–5. IEEE.

Eunji Lee, Sihyeon Kim, Sundong Kim, Soyeon Jung, Heeja Kim, and Meeyoung Cha. 2024. Explainable product classification for customs. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–24.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Saikat Mukherjee, Dmitriy Fradkin, and Michael Roth. 2008. Classifying spend descriptions with off-the-shelf learning components. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, pages 53–60. IEEE.

Zaruhi Navasardyan. 2024. INTERPRETABLE AND GENERALIZABLE HTS CODE CLASSIFICATION FRAMEWORK. *Economics, Finance and Accounting*, 1(13):140.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report.

Awdhesh Kumar Singh and Rajendra Sahu. 2004. Decision support system for HS classification of commodities. In *Proceedings of the 7th International Conference on Computer and Information Technology*.

Margarita Spichakova and Hele-Mai Haav. 2020. Application of machine learning for assessment of HS code correctness. *Baltic Journal of Modern Computing*, 8(4).

Haichao Sun, Chengjie Zhou, and Chao Che. 2025. Customs commodity classification method based on the fusion of text sequence and graph information. *Expert Systems*, 42(6):e70057.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Shijie Wang, Wenqi Fan, Yue Feng, Lin Shanru, Xinyu Ma, Shuaiqiang Wang, and Dawei Yin. 2025. Knowledge graph retrieval-augmented generation for llm-based recommendation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27152–27168.

World Customs Organization. 2022. *Harmonized Commodity Description and Coding System*. World Customs Organization, Brussels.

Ziyao Zhang, Chong Wang, Yanlin Wang, Ensheng Shi, Yuchi Ma, Wanjun Zhong, Jiachi Chen, Mingzhi Mao, and Zibin Zheng. 2025. Llm hallucinations in practical code generation: Phenomena, mechanism, and mitigation. *Proceedings of the ACM on Software Engineering*, 2(ISSTA):481–503.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. 2023. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36:50117–50143.

A Prompt Templates

For reproducibility, we include polished versions of the two prompts used in the online reasoning loop. Unlike the shortened sketches in the main text, the templates below closely match the production prompts instantiated during inference, while replacing runtime values with placeholders.

Selection prompt. This prompt is called at every hierarchy level to select the next-hop node from the current node's valid *Contains* children. Its main role is to enforce local, level-wise decisions under the candidate set $\mathcal{C}(v_i)$ defined in Section 3.4. Accordingly, the prompt normally permits selection only among valid child nodes of the current node, while also allowing a controlled return to the previous node when no child yields a defensible continuation.

> You are an experienced customs HS classification expert. Your task is to traverse the HS knowledge graph level by level according to the product description.
>
> Product description: [PRODUCT_DESCRIPTION]
>
> Visited path: [CURRENT_PATH]
>
> Current node: [CURRENT_NODE_NAME] ([CURRENT_NODE_ID])
>
> Candidate child nodes under the current *Contains* relation: [CHILD_OPTIONS]
>
> Instructions:
>
> 1. Select exactly one next-hop node from the candidate child nodes listed above. The selected node must be one of the provided candidates, unless you determine that none of them can plausibly extend the current path and a return to the previous node is necessary.
>
> 2. Base the decision on the product description, the current reasoning path, and the node summaries. Prefer the child node whose scope is most semantically consistent with the product while remaining compatible with the top-down HS hierarchy.
>
> 3. Do not jump across levels or propose nodes outside the candidate set. If the current node cannot be extended by any listed child node after careful comparison, you may return to the immediately previous node in the visited path by outputting its node ID.
>
> 4. Backtracking is a last-resort action. Use it only when all listed child nodes are clearly incompatible with the product or would lead to an unstable continuation.
>
> 5. Regulatory exclusion and redirection are handled in a later step. At this stage, focus on either selecting the locally valid child node that best extends the current path or returning to the previous valid node.
>
> 6. The output must be an HTML fragment in the following format, and the node ID must not contain periods:
>
> `<reason>brief reasoning</reason>`
>
> `<select>NODE_ID</select>`
>
> 7. Do not output anything else.

At runtime, each candidate in `[CHILD_OPTIONS]` is expanded as:

> `code:"NODE_ID" | name:"NODE_NAME"`
> Summary: [NODE_SUMMARY]

Redirect prompt. This prompt is triggered when the current node has outgoing exclusion or redirection rules. Its purpose is to implement the regulatory correction step described in Section 3.4, namely deciding whether the currently extended path should be kept or redirected to a compliant alternative.

> Product description: [PRODUCT_DESCRIPTION]
>
> Current node: [CURRENT_NODE_NAME] ([CURRENT_NODE_ID])
>
> Visited path: [CURRENT_PATH]
>
> According to the redirect rules, determine whether the path should be redirected. Candidate options: [REDIRECT_OPTIONS]
>
> If the current path remains legally valid, output `none`.
>
> Rules:
>
> 1. Evaluate whether the current path violates any exclusion or redirection condition associated with the current node. Redirect only when a listed rule is clearly triggered by the product description.
>
> 2. The decision should consider the product's essential characteristics, including function, material, processing state, and other attributes explicitly mentioned in the rule text or node summary.
>
> 3. If redirection is required, select one target node from the listed redirect options only. Do not invent a new destination outside the provided options.
>
> 4. Redirection operates as a path-level correction step. If no listed rule invalidates the current path, keep the current path and output `none`.
>
> 5. If all listed redirection targets would still leave the path non-compliant or clearly unsuitable, you may fall back to the immediately previous node in the visited path by outputting that node ID.
>
> 6. Consider the visited path to avoid repeated redirection loops. Do not redirect to a target that would recreate an already rejected path.
>
> 7. The output must follow the format below:
>
> `<reason>brief reason</reason>`
>
> `<redirect>NODE_ID/none</redirect>`
>
> 8. Do not output anything else.

At runtime, each candidate in `[REDIRECT_OPTIONS]` is expanded as:

```
Redirect to TARGET_NODE_ID | TARGET_NODE_NAME
Condition: [REDIRECT_CONDITION]
Summary: [TARGET_NODE_SUMMARY]
```

In implementation, the selection prompt is responsible for locally constrained child-node selection with limited backtracking, whereas the redirect prompt is responsible for rule-triggered path correction and fallback when no compliant redirection remains. This separation matches the Select–Redirect mechanism illustrated in Figure 4 and Algorithm 1.