scientific data

OPEN

DATA DESCRIPTOR

Business Intent and Network Slicing Correlation Dataset from Data-Driven Perspective

Jie Li[1], Sai Zou[1]✉, Yanglong Sun[2], Hongfeng Gao[1] & Wei Ni[3]

Intent-Based Networking (IBN) is an emerging network management technology that enables automated configurations based on user intents. A critical aspect of IBN is the accurate and autonomous extraction of user intents and their translation into a language comprehensible by network management systems. However, the current scarcity of publicly available datasets for intent extraction presents significant challenges. With the rise of big data, data-driven research methods for investigating future networks have become a trend. This paper presents a Business Intent and Network Slicing Correlation Dataset (BINS) to advance research in next-generation networks. The dataset includes user business intent descriptions, annotated intent data, and correlations between business intents and network slices. We utilize natural language processing techniques based on named entity recognition and third-party data analysis tools such as DataProfiler to validate the data quality of BINS, confirming its reliability. As a cutting-edge dataset for network intent recognition, BINS will contribute to the development of IBN systems and provide valuable data resources for researchers and practitioners exploring application interactions and related technologies.

Background & Summary

Future networks will face increasingly complex and diverse demands and challenges with the rapid development of emerging technologies such as the Internet of Things (IoT), cloud computing, and big data[1]. In this context, it is essential for network management systems to embrace automation and intelligence to ultimately achieve closed-loop autonomy[2], which would enable self-learning and evolution capabilities. However, without sufficient data support, closed-loop autonomous systems would struggle to make effective decisions and adjustments. Data serves as the cornerstone of system operation, providing real-time network state information and historical operational records, which are crucial for enabling systems to self-optimize and self-protect[3]. The core value of data-driven approaches lies in leveraging in-depth data analytics to inform decisions and actions, and closed-loop autonomous systems serve as a key platform for realizing this value proposition. By integrating data analytical insights into actual network operations, closed-loop autonomous systems can automatically execute adjustments and optimization actions, thereby fully demonstrating the advantages of data-driven approaches[4]. Thus, data-driven network reform is the necessary pathway to addressing network complexity, optimizing resource allocation, enhancing network security, promoting business innovation, and achieving network intelligence[5]. As technology continues to advance and application scenarios expand, data-driven network reform will play an increasingly pivotal role in the future.

In the process of intent-based network management, users or administrators typically express specific requirements regarding network performance, quality of service, and resource allocation through natural language[6]. These requirements are first parsed by a business intent recognition system and translated into machine-understandable business intent data. Based on this, the network management system orchestrates resources and generates corresponding network slices to flexibly respond to different business needs[7]. Throughout this process, the network management system relies on intent data and the associated data to link business intent to network slices, and achieve end-to-end service management from demand to delivery[8]. First, an appropriate entity set is needed for intent recognition, which is fundamental to the effective operation of intent-based network management systems. Business intent extraction is typically achieved using

[1]College of Big Data and Information Engineering, Guizhou University, Gui Yang, 550000, China. [2]College of Navigation, Jimei University, Xiamen, 361000, China. [3]CSIRO, Data61, Sydney, 2000, Australia. ✉e-mail: dr-zousai@foxmail.com

natural language processing[9] techniques, such as BiLSTM[10] (bidirectional long short-term memory) model or by fine-tuning Google's BERT[11] (Bidirectional Encoder Representations from Transformers) model. To enhance the accuracy and generalization capability of these models, it is essential to design entity sets tailored to specific tasks. Second, labeled business intent description data is required. Most existing network and business datasets describe only one aspect of characteristics, making it difficult to adapt to the rapidly evolving network scenarios. This leads to a shortage of training data for intent recognition systems. Directly obtaining research data from relevant institutions is often difficult, as these organizations are typically reluctant to share such data[12], and such data are rarely collected and organized. Furthermore, public network datasets are rendered unreliable and sometimes outdated due to the rapid changes in network scenarios and patterns[13]. Finally, the generation and optimization of network slice data should be dynamically adjusted based on business requirements. However, current network slice generation still predominantly relies on standard service models based on network communication functions, which involve selecting appropriate slices from pre-existing ones. To achieve more flexible and business-driven network slice orchestration, it is essential to construct intent datasets that can dynamically generate and optimize slice data according to business needs.

In this paper, we create a new Business Intent and Network Slicing dataset (BINS) to bridge the gap between the agent need for such a dataset and the lack of the dataset. The BINS dataset provides richly structured and meticulously curated descriptive intent data for intent-based networking (IBN)[14], with its data sourced from three primary channels. The first data source comes from China Telecom Sichuan Branch (Sichuan Telecom), consisting of real network operational data monitored and recorded within their intent-based network. The second part of the dataset is generated through manual construction by professional network engineers and by professionals simulating network operators expressing network intents. The third part of the BINS dataset is derived from parsing relevant academic papers, industry standards, and publicly available network documents from official websites of universities and research institutions, enterprises, and telecom operators. Within these network scenarios, we encompass multiple application contexts, including campus environments, industrial intelligent networking under the Industrial Internet, and real-time monitoring, covering over 200 distinct network intents. In this context, "intent" is defined as high-level goals or requirements expressed by network stakeholders, indicating the anticipated services, performance, and behavior of the network. By manually annotating these intent entities and relationships, the final dataset contains more than 100,000 intent entities and over 40,000 unique triples. These triples can be used to construct knowledge graphs[15] for intent recognition, providing critical support for further development of IBN systems.

The primary objective of the BINS dataset is to advance the development of IBN, which is crucial for the applications of fully autonomous networks[16] in the future. This is particularly important for emerging network scenarios stemming from various technologies, such as smart environments[17], IoT[18], vehicular networks (V2X)[19], healthcare[20], and next-generation 6G networks[21]. In these contexts, intelligent and advanced automation software mechanisms replace error-prone manual network configurations, thereby enhancing the efficiency of network configuration and reducing costs. Furthermore, the BINS dataset holds significant potential for broader applications, such as information retrieval and application interaction in natural language processing. In these fields, using intent datasets to train and evaluate natural language understanding models can effectively identify user requests and commands, enabling the provision of relevant information or actions and improving the overall user experience. Overall, the BINS dataset not only offers vital support for intent recognition tasks in closed-loop autonomous systems but also serves as a valuable resource for the intelligent development of other applications across various domains.

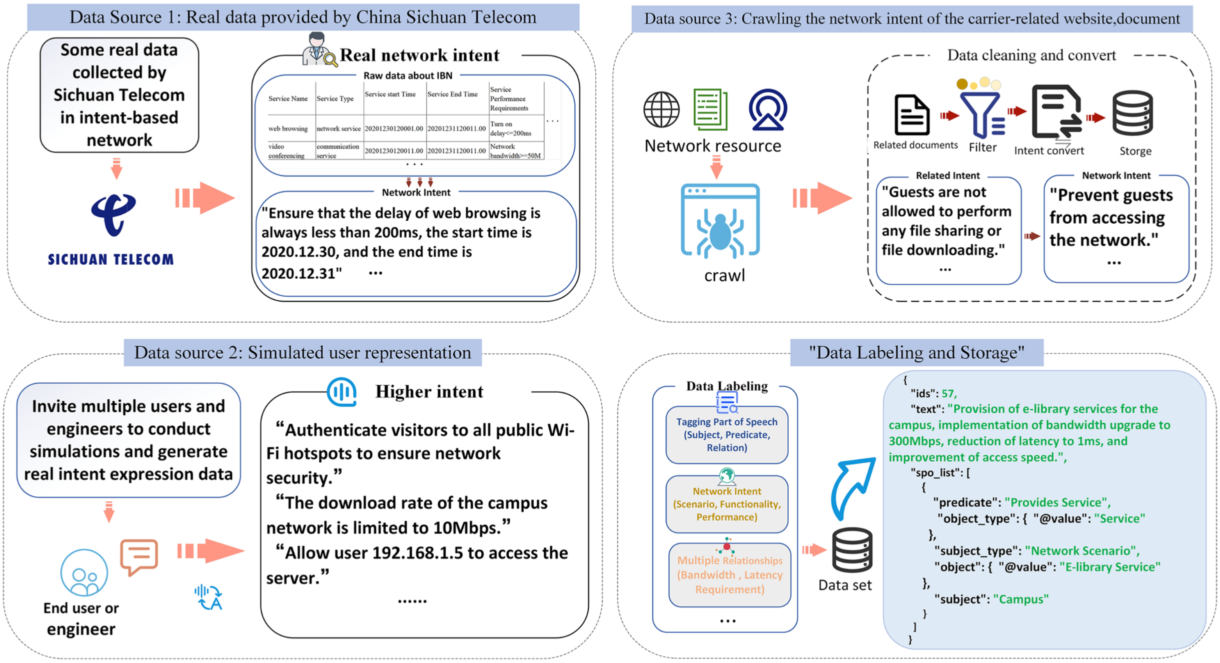
Methods

This section outlines the entire process of data generation, covering the various data sources, the data processing workflows and annotation methods, and providing an explanation of the included label categories and their definition methods.

Data collection methods. Figure 1 illustrates the three data sources of BINS and their corresponding data preprocessing processes used in this study. The data sources include:

Data source 1. As shown in Fig. 1, the first source of data comes from real-world operational data collected during intent-based network operations provided by Sichuan Telecom, Sichuan, China. This data was monitored and recorded within the intent-based network, covers network demands and performance metrics across various business scenarios, with a total of 10,001 records. The data metrics include service name, service type, start and end times, and network performance requirements. Based on these metrics, the data can be transformed into corresponding expressions of network intent. For instance, a record with a service name "web browsing", service type "network service", start time "20201230", end time "20201231", and a performance requirement of "latency $\leq$200 ms" is translated into a user intent expression such as: "Ensure that the latency for web browsing is $\leq$200 ms from 20201230 to 20201231" or "Maintain web browsing latency consistently $\leq$200 ms". Through such transformations, irrelevant data was filtered out, yielding over 14,000 intent expressions from real business scenarios as part of the BINS dataset.

Data source 2. As illustrated in Fig. 1, some data was manually generated by network engineers based on their practical work experience. These intent data samples were derived from common scenarios in network configuration, troubleshooting, and business requests. Since these network engineers possess specialized knowledge, the intents they created are more detailed than those expressed by end users, often including more specific network parameters, such as bandwidth and latency, which are particularly beneficial for building intent-based network systems.


Data Source 1: Real data provided by China Sichuan Telecom
Some real data collected by Sichuan Telecom in intent-based network
SICHUAN TELECOM
Real network intent
Raw data about IBN
Network Intent
Data source 3: Crawling the network intent of the carrier-related website,document
Network resource
crawl
Data cleaning and convert
Related documents
Filter
Intent convert
Storge
Related Intent
Network Intent
Data source 2: Simulated user representation
Invite multiple users and engineers to conduct simulations and generate real intent expression data
End user or engineer
Higher intent
"Data Labeling and Storage"
Data Labeling
Tagging Part of Speech (Subject, Predicate, Relation)
Network Intent (Scenario, Functionality, Performance)
Multiple Relationships (Bandwidth , Latency Requirement)
...
Data set

Service Name	Service Type	Service start Time	Service End Time	Service Performance Requirements
web browsing	network service	20201230120001.00	20201231120011.00	Turn on delay<=200ms
video conferencing	communication service	20201230120011.00	20201231120011.00	Network bandwidth>=50M

"Ensure that the delay of web browsing is always less than 200ms, the start time is 2020.12.30, and the end time is 2020.12.31" ...

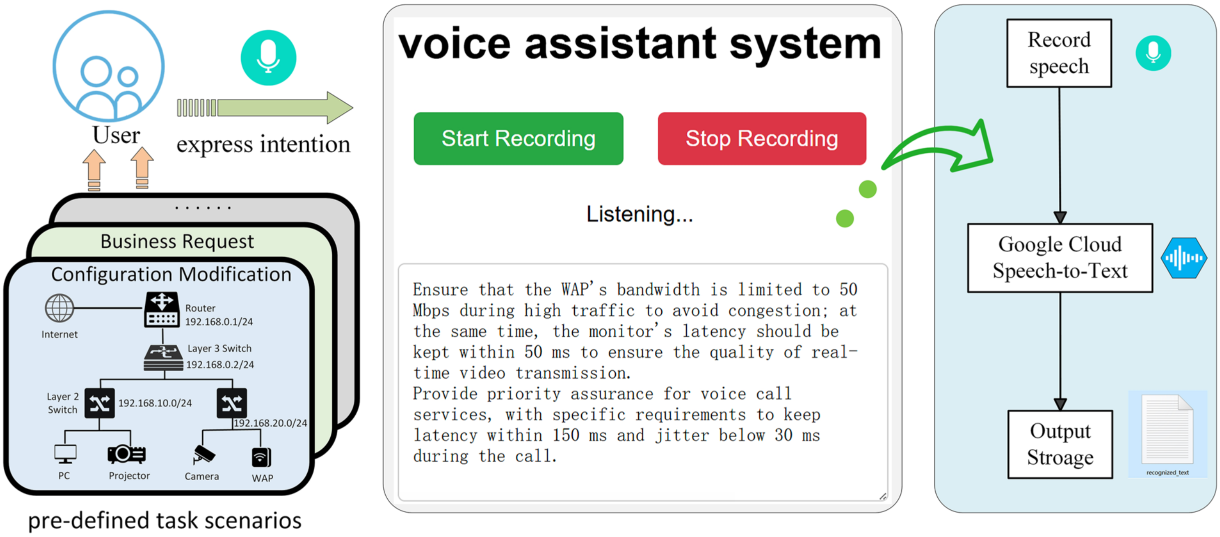
"Guests are not allowed to perform any file sharing or file downloading." ...

"Prevent guests from accessing the network." ...

"Authenticate visitors to all public Wi-Fi hotspots to ensure network security."
"The download rate of the campus network is limited to 10Mbps."
"Allow user 192.168.1.5 to access the server."
......

```
{
  "ids": 57,
  "text": "Provision of e-library services for the campus, implementation of bandwidth upgrade to 300Mbps, reduction of latency to 1ms, and improvement of access speed.",
  "spo_list": [
    {
      "predicate": "Provides Service",
       "object_type": {  "@value": "Service"
      },
       "subject_type": "Network Scenario",
       "object": {  "@value": "E-library Service"
    },
       "subject": "Campus"
    }
  ]
}
```

Fig. 1 The three sources of data, along with the data preprocessing and annotation process, all three data sources undergo corresponding annotation processing, with an annotation example provided in the lower-right corner.



Ensure that the WAP's bandwidth is limited to 50 Mbps during high traffic to avoid congestion; at the same time, the monitor's latency should be kept within 50 ms to ensure the quality of real-time video transmission.
Provide priority assurance for voice call services, with specific requirements to keep latency within 150 ms and jitter below 30 ms during the call.

Fig. 2 An example of the intent collection API, where the simulator interacts with the voice assistant system in a predefined task scenario, generating real business intent expression data through voice input.

Volunteers were invited to simulate user network requests, generating user intent data through interaction with a voice assistant system. This process was primarily facilitated through a voice interface, where volunteers engaged with pre-defined task scenarios, such as network configuration modifications, business requests, and performance optimization, using voice inputs. The user interface of the voice assistant was developed using HTML, CSS, and JavaScript, and incorporates the Google speech recognition API Google Cloud Speech-to-Text[22]. This API is renowned for its high accuracy and real-time processing capabilities, capable of rapidly and accurately converting spoken input into text, ensuring the smoothness and accuracy of the data collection process. Figure 2 provides an example of this process.

On the other hand, different intent expression paradigms can significantly impact the accuracy of intent recognition systems and the subsequent generation of intent strategies. Therefore, formalizing the grammar of intent expressions is crucial[23]. Additionally, Some simulated users may be limited by their limited network expertise, which could result in ineffective intent expressions. To more accurately simulate end-user network requests, this paper refers to the intent classification provided in IETF RFC9316[24] and introduces a reference

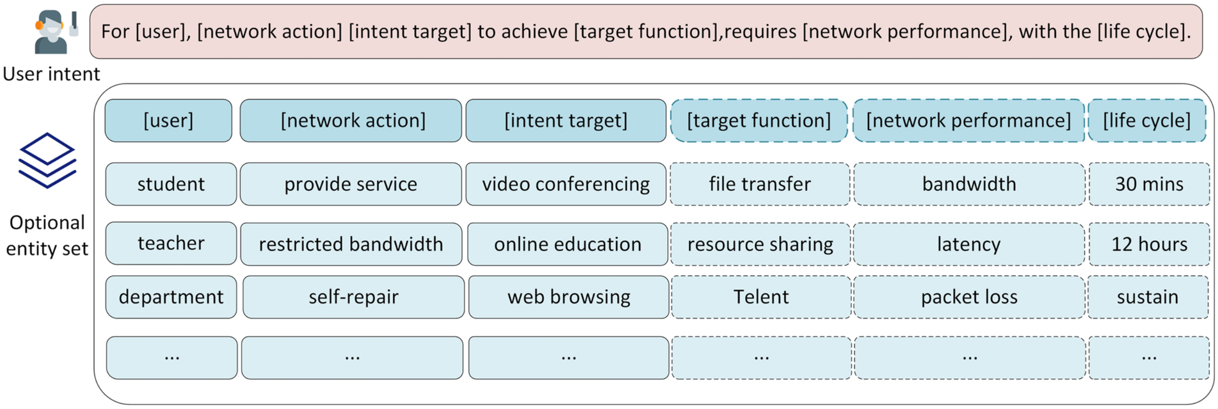


Fig. 3 The reference model provided for standardizing intent expression allows participants to select and combine relevant entity labels based on their needs, forming key entity groups to accurately express business intent.

model for intent expressions, simulated users limited by network knowledge can refer to this template, as shown in Fig. 3. This model standardizes intent expressions and is composed of the following entity label sets:

- Intent user (customer, or network or service operators). The "user" entity is the declarer of the intent, and can include technical users such as network experts, network administrators, service providers, or non-technical end users, such as teachers and students in a school, or administrative departments in an enterprise.
- Network action. This entity represents the manner in which the required service or the network should achieve a certain state, such as providing a service, limiting bandwidth, or self-healing.
- Intent target. This entity collects the user's needs, services, or objectives that need to be fulfilled, such as web browsing, online education, or video conferencing.
- Target device or function. This entity denotes the network function (e.g., data communication, resource sharing) or physical device (e.g., routers, switches) used to deploy the intent. This is an optional entity.
- Network performance. This entity collects the desired network performance metrics specified by the user, such as bandwidth, latency limitations, jitter requirements, etc. This is also an optional entity.
- Lifecycle (persistent, transient). This entity refers to the duration of the intent, which is an optional entity and can range from minutes to hours or longer.

It is noted that the standardized intent expression process illustrated in Fig. 3 serves as a reference guide only and does not require strict adherence in practical applications. Participants may flexibly adjust the sequence of entity label usage and selectively employ partial or complete entity label sets according to specific application scenarios and practical requirements, thereby achieving diversified and effective intent expression. For instance, in a campus network scenario, a network administrator may express the following intent: "For students, providing web browsing services to achieve resource sharing, requires network bandwidth exceeding 100Mbps with the service being available for a minimum of 6 hours." This can also be simplified to: "For students, providing web browsing services to achieve resource sharing." The corresponding entity labels are: [user]: "student", [network action]: "provide service", [intent target]: "web browsing", [target function]: "resource sharing", [network performance]: "bandwidth exceeding 100Mbps", [lifecycle]: "6 hours". In this case, [target function], [network performance], and [lifecycle] are optional entities.

Data source 3. As shown in Fig. 1, network intents were parsed by analyzing academic papers, industrial standards, and websites related to operators, educational institutions, and enterprises. Telecom operators and technical support websites typically contain a wealth of information, including network issues raised by researchers, intent examples, and corresponding solutions. These data encompass diverse types of intents and data. Data is collected through methods such as keyword search, topic search, and the snowball method. Subsequently, noise is removed and irrelevant information is filtered manually, while PySpellChecker[25] is used for spell checking to ensure high data quality. Subsequently, through a combination of automated tools and manual review, this data was transformed into valid intent expression data. For example, in a journal[26] case study on IBN, the authors presented the following intent: "Transfer a common-level video service from user A in Beijing to user B in Nanjing." After being formatted, this intent was passed on to the intelligent policy mapping module, which parsed the intent and broke it down into a specific service function chain (SFC), such as network address translation (NAT) and firewall functions, then constructed the corresponding SFC request. In this case, user A and user B can be represented by their respective IP addresses. Hence, the intent could be translated into: "Transfer a common-level video service from user 220.15.2.10 in Beijing to user 210.59.4.15 in Nanjing" (with assumed IP addresses). Through this process of data collection and transformation, a total of 274 intent entries were gathered from relevant documents and websites. After effective conversion, 257 verified intent data entries were obtained.

Classification	Type	Numbers	Percentage
Age	21–25	8	54%
	25–30	2	14%
	>30	5	32%
Occupation	Graduate or Undergraduate	8	53%
	Junior Network Technician	2	14%
	IT Support Specialist	2	14%
	Senior Network Engineer	3	19%
Years Working Experience	<1 years	9	60%
	1–5 years	2	13%
	>5 years	4	27%

Table 1. The distribution of participants used to illustrate compliance and ensure data quality.

Data processing and annotation. To ensure data accuracy and reliability, targeted preprocessing measures were implemented based on different data sources. For the real-world data provided by Sichuan Telecom, consistency and data completeness were manually verified, irrelevant network metrics were removed, and the data was transformed into structured intent expressions. For the data constructed by network engineers, manual reviews were conducted to ensure data relevance and reliability, guaranteeing that the constructed data accurately reflected actual network operational demands and intents. Data generated by simulated users was transcribed into text via the automated voice assistant system, after which it underwent redundancy and noise checks, as well as language normalization. Due to potential inaccuracies or irrelevant content in the speech recognition process, careful manual filtering of valid intent data was required. For network-parsed data, the processing was more complex, involving manual filtering, redundancy removal, noise reduction, and extraction of content related to network intents, which were then converted into structured intent descriptions to ensure data relevance and usability.

During network data processing, raw data collected from multiple sources could be affected by various distortions, including semantic redundancy, non-ASCII encoded characters, spelling errors, and unrelated expressions. These distortions are categorized as follows:

- Semantic repetition. Sentences that convey the same meaning but differ in linguistic expression. For instance, "Ensure that there is no buffering while playing high-definition video" and "Ensure stable network performance during HD video streaming without any delays" express the same intent, which is to guarantee a stable and buffer-free network connection while watching HD videos. Although the wording differs, the core intent is the same: ensuring smooth video playback of streaming media videos.
- Non-ASCII encoding. Sentences containing special characters, such as Greek letters, mathematical symbols, or consecutive punctuation marks.
- Spelling errors. To ensure linguistic accuracy, an automated Python-based spell-checking tool, PySpellChecker[25] was used to detect and correct spelling errors within the data.
- Unrelated expressions. These are statements that contain additional information unrelated to the core intent. For example, in the sentence, "The company has recently expanded its workforce, so we need to ensure that video conferencing for all employees has no network latency," the unrelated portion, "The company has recently expanded its workforce," should be removed, leaving the key intent as "Ensure no network latency for video conferencing."

All preprocessed data underwent entity annotation, relationship annotation, and slice type annotation to make it suitable for natural language processing tasks, such as Named Entity Recognition (NER)[27] and relation extraction. Entity labels are used to identify key elements in the text, such as business types, network performance, or functionalities; relationship labels describe the interconnections between business entities, such as "provides" or "ensures"; slice type labels specify the corresponding network slice type for the requirements expressed in the intent, based on domain-specific needs. These slice types include eMBB, URLLC, and mMTC, which are used to differentiate various network service types. To further enhance the utility of the data, the BIO (Begin, Inside, Outside)[28] annotation scheme was applied to label the raw data, where the BIO annotation is applicable to tokenized text. The BIO annotation scheme is widely used in sequence labeling tasks, such as NER, tokenization, and syntactic analysis. It assigns a label to each word to indicate its entity type, with "B" indicating the beginning of an entity, "I" the middle part, and "O" representing non-entity words. The annotated structured data was stored in JSON format, ensuring consistency and usability, and providing a high-quality data foundation for the subsequent training and deployment of intent recognition models.

Data Records

The dataset files are publicly accessible through the data platform figshare[29]. The data is organized into three folders. The raw data collected is stored in the CSV format, while the annotated and structured data is stored in the JSON format.

Personnel and user interface description folder. volunteer.csv is an informational file recording the basic details of the volunteers who participated in the experiment, including their ID, age, and occupation. Table 1 presents the distribution of volunteers. The files "App.js", "index.html" and "styles.css" form the user interface used

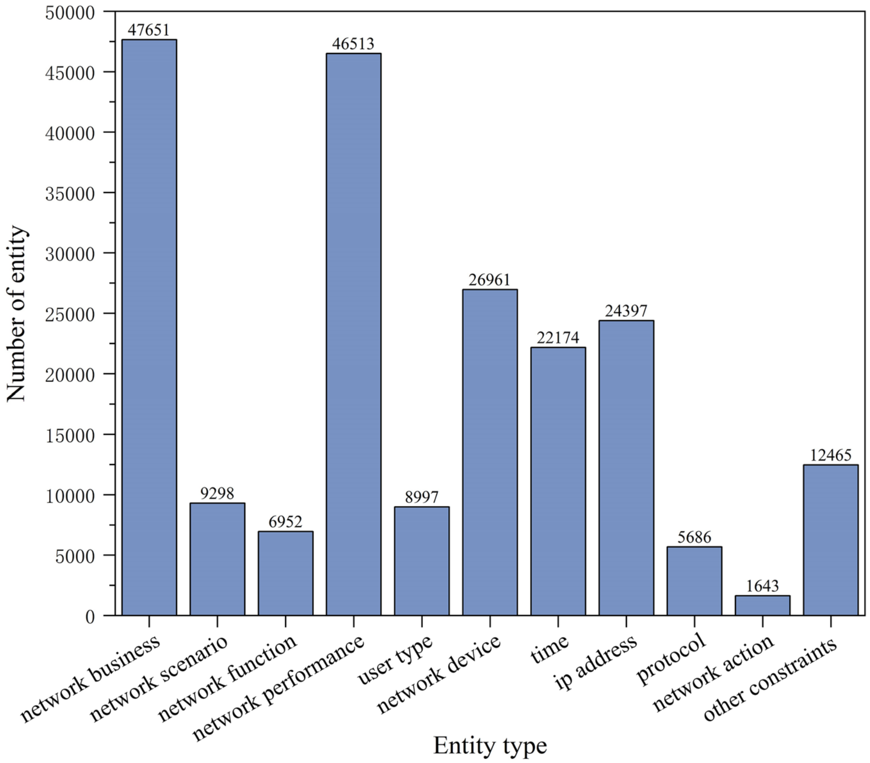


Fig. 4 The statistical information on the number of major entity categories included in the BINS dataset.

Data Source	Entity Counts	Unique Relationships Counts	Slice Types Counts	Unique Triplets	Avg Span (Entities)	Avg Span (Relationships)
sichuan.json	80,792	48	14,968	17,725	11.7	16.5
engin.json	31,716	65	5,338	8,070	11.6	15.1
volu.json	97,323	70	16,367	19,085	11.2	15.2
docu.json	2,906	21	201	1,443	13.4	17.6

Table 2. Statistical information of various labels from different data sources.

for volunteers to collect network intent data. These files are developed using HTML, CSS, and JavaScript, and the integrated with the Google Cloud Speech-to-Text API to convert the network intents expressed by volunteers into text (source code is detailed in the "Code Availability" section).

Raw intent data folder. This folder comprises multiple files containing preprocessed data from various sources. Each record includes user ID, intent description, and creation time. For simulated or engineer-constructed data, refer to the files "engin_ori.csv" and "volu_api.csv". For document-parsed data, including network requirements, source URL, and collection times, refer to the files "docu.csv" and "sichuan.csv".

Labeled intent data folder. After processing the raw data as described earlier, the annotated data is stored in four JSON files: "engin.json," "volu.json," "docu.json," and "sichuan.json". Each record in these files consists of text, entity type, entity, relationship, slicing type and the BIO annotations of entities. These annotated structured data files are not only used for training intent recognition models, but also for common natural language processing tasks, such as entity extraction and relationship extraction. Additionally, each text in the files is labeled with a BIO sequence based on the position of its entities within the text, making it more suitable for training intent extraction models. Figure 4 shows the statistical information on the number of major entity categories included in the BINS dataset. Table 2 lists the total number of entity, unique relationships, and slice types contained in each file. It also includes the average span length of entities and relationships, as well as the count of unique triples for each data source.

Technical Validation

To ensure the availability and assess the reliability of the data for network intent recognition, the training samples should comprehensively cover the network domain as much as possible, enabling the efficient identification of business intents within networks. To achieve this, it is essential to ensure that the data has been preprocessed, and is consistent and complete in order to assess its reliability. Data preprocessing is critical for reducing the

Main Metrics	Metric Description	Value
Data Correlation	The format of the input data set is consistent	100%
Data Repeatability	The number of data non-duplicates accounts for the total number of data	98%
Data Completeness	The proportion of rows that are fully comprised of no null values (null rows) to the total number of rows	98%

Table 3. The main quality indicator descriptions and corresponding values of the adopted DataProfiler.

impact of data noise and irregularities on model training, which can otherwise affect the reliability analysis of the data. This aspect has already been described in the data processing subsection of the methods section.

Data consistency refers to whether the data follows a unified standard and maintains a uniform format[30]. Initially, after the annotation of entity categories and relationships, the data is stored in a structured manner to facilitate reuse. Additionally, each text is labeled in the BIO format, making it suitable for the primary task or other natural language processing tasks.

Other aspects of detection utilized Capital One's DataProfiler[31], a powerful, flexible, and easy-to-use tool for data quality and metadata analysis. It is capable of automatically identifying file types, analyzing data quality, and generating easily interpretable visual reports. The main quality metrics used are described in Table 3.

To validate whether this dataset can be used to train intent recognition models and to assess the performance of these models, this paper employs the named entity recognition method based on BERT[11] to verify if the data can be applied in intent recognition scenarios. By utilizing NER to extract key entities and related parameters from unstructured text, the model can effectively recognize users' network intents. BERT is a pre-trained model introduced by Google in 2018, which includes 12 Transformer blocks, each equipped with multi-head self-attention mechanisms and feedforward neural network layers, make it particularly effective in natural language processing tasks. In the experiments, for comparison, the BERT-CRF[32] and BERT-BiLSTM-CRF[33] models were used to train the intent recognition system. The BERT-CRF model adds a Conditional Random Field (CRF) layer on top of the BERT output to capture dependencies between labels. Specifically, given the BERT output $\overrightarrow{H}$ for the input sequence, the CRF layer calculates the score $s(X,\ Y)$ for each possible label sequence using the following scoring function:

$$s(X,\ Y) = \sum_{i=1}^{n} W_{y_{i-1}, y_i} + \sum_{i=1}^{n} P_{y_i} \cdot h_i,$$

where w_{y_{i-1}, y_i} is the transition score between labels, P_{y_i} is the feature weight matrix, and h_i is the BERT output. The BERT-BiLSTM-CRF model further introduces a BiLSTM layer between the BERT output and the CRF layer. The BiLSTM layer captures bidirectional contextual information of the input sequence through forward and backward LSTM units, with its output H' serving as the input to the CRF layer: $H' = \left[H_f;\ H_b\right]$. All BERT-based models are trained using the Adam optimizer, with the learning rate set to 5×10^{-5}. The output layer of the models uses Softmax as the activation function, and the model loss is computed using cross-entropy loss. Additionally, we experiment with fine-tuning Spacy (https://spacy.io/) NER models over the pre-trained DeBERTa-v3-large MLM[34] to quantify the performance of the BINS dataset on the latest model. DeBERTa's powerful pre-training capabilities help capture contextual information effectively, thereby enhancing entity recognition in complex contexts. Spacy NER is a deep learning-based tool designed to identify and classify entities from text. It utilizes CNN, RNN, or Transformer architectures, and incorporates a CRF layer to optimize the prediction of entity labels. Spacy NER supports various entity types and offers pre-trained models in multiple languages. The models are trained on large-scale annotated corpora, providing excellent out-of-the-box performance, while also supporting customization for domain-specific training needs. Spacy NER is widely applied in fields, such as information extraction, knowledge graph construction, and chatbot development. By fine-tuning Spacy NER over the DeBERTa MLM, the model's entity recognition capabilities in IBN scenarios can be enhanced, thereby validating the generalization ability of the data.

To evaluate the performance of the models, precision, recall, and F1-score[35] were selected as evaluation metrics, which are commonly used for named entity recognition tasks. Precision refers to the proportion of true positive samples among those predicted as positive by the model. Recall indicates the proportion of actual positive samples that were correctly predicted by the model. F1-score considers the harmonic mean of precision and recall, providing a balanced assessment of both. When precision and recall are both high, the F1-score are high. When evaluating F-scores, precision, and recall, exact matching[36] was used. This is because in the IBN systems, there is often a strict correspondence between entities and categories. Using exact matching ensures the evaluation results accurately reflect the model's performance under stringent conditions. Exact matching measures the model's adaptability to the datasets, avoiding biases in generalization ability and data quality that may arise from relaxed matching methods. In the experiment, the labeled "engin.json" dataset was used as the foundation to quantify the performance of the model. This dataset contains 14,466 entries, divided into 80% training data and 20% testing data. To observe the convergence of the models during training, the performance of both models was plotted in Fig. 5. The training steps represent the ratio of the number of training samples to the batch size, multiplied by the number of training epochs. In the experiments, the batch size was set to 8, and the epoch was set to 1. As shown in Fig. 5, the BERT-BiLSTM-CRF model converged at 1,000 training steps with an F1-score of approximately 93%, while the BERT-CRF model converged at 1,200 training steps with an F1-score of 90%. In contrast, the fine-tuned DeBERTa-v3-large MLM model achieved convergence at 1200 training steps, with an F1 score of 94%.

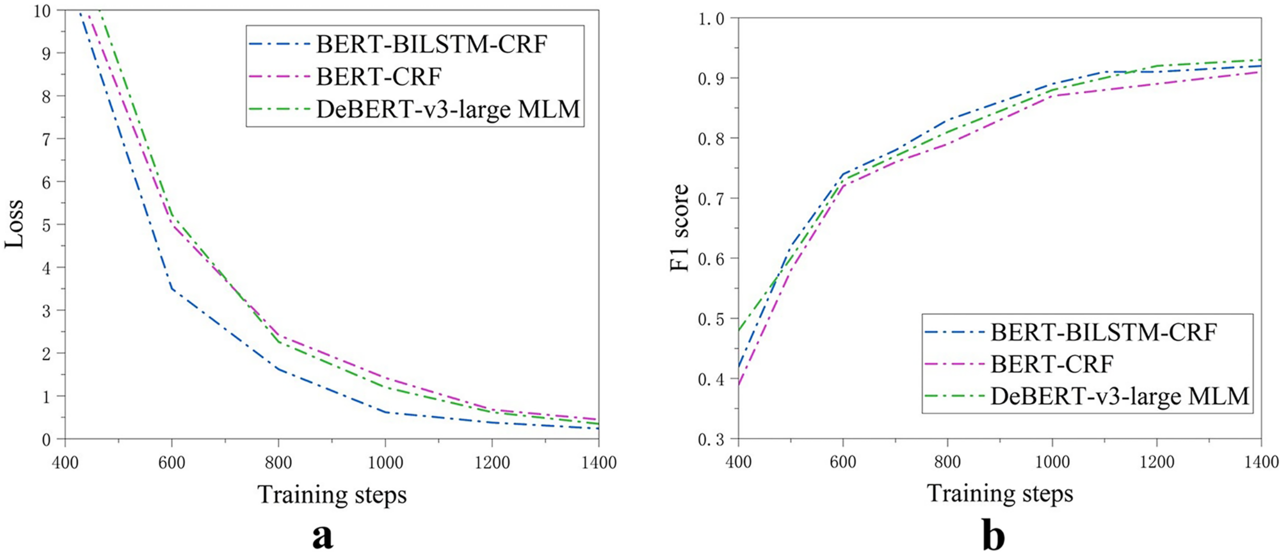


Fig. 5 The performance of the three models on the "engin.json" dataset is shown. (**a**) Presents the convergence of each model with the number of training steps, while (**b**) illustrates the change in the F1 score of each model throughout the training process.

model	"sichuan.json"			"engin.json"			"volu & docu.json"		
	P	R	F1	P	R	F1	P	R	F1
BERT-CRF	0.90	0.91	0.91	0.90	0.89	0.90	0.86	0.85	0.85
BERT-BiLSTM-CRF	0.93	0.92	0.93	0.93	0.92	0.92	0.89	0.88	0.89
DeBERT-v3-large MLM	0.94	0.95	0.94	0.93	0.95	0.94	0.91	0.93	0.92

Table 4. The performance of the three different models across various data sources.

To evaluate the contribution of the BINS dataset to model generalization, we used the remaining three datasets with 80% allocated for the training data and 20% for the testing data in the experiments. Due to the smaller sample size of the "docu.json" dataset, it was combined with the "volu.json" dataset and used together in the experiments. Table 4 shows the performance of the three models upon convergence across various data sources. The BERT-BiLSTM-CRF model outperforms the BERT-CRF model of all metrics and convergence speed on all datasets, indicating that adding a BiLSTM layer to the BERT model helps capture richer contextual information, thereby improving the named entity recognition performance. Although the BERT-BiLSTM-CRF model performs excellently on BINS dataset, the DeBERTa-v3-large MLM model, with its more efficient long-range dependency capturing capability and context processing mechanism, can better adapt to different domains and tasks. It excels particularly in handling complex or variable entity annotations. As a result, the DeBERTa-v3-large model significantly outperforms the BERT-BiLSTM-CRF model across all datasets.

The experiments demonstrated that the labeled dataset used in model training and testing performed well, indicating its suitability for intent recognition tasks. This also validates the dataset's effectiveness and applicability, providing a high-quality data foundation for building efficient network intent recognition models.

Usage Notes

The collected intent-based network dataset in this paper can be used to train and optimize intent recognition systems. The system precisely determines users' network intent by interpreting and analyzing their natural language input. It then translates this meaning into appropriate network strategies, including configuration, troubleshooting, and performance improvement. For example, in terms of network configuration, when a user inputs commands like "configure VLAN" or "set firewall rules", the system can efficiently interpret the request and convert it into the appropriate network policy. After validation, the system generates the corresponding network slice and deploys it to the relevant network infrastructure, thereby enhancing the automation and intelligence of the network management system.

Additionally, these datasets can be extended to applications in other fields, such as information retrieval and application interaction. In information retrieval, users' queries often carry explicit intent. By training intent recognition models using our constructed datasets, search engines can better understand the intent behind user queries, thereby providing more relevant and precise search results. For example, when a user queries "how to improve network speed," the intent recognition model can interpret that the user is seeking better network performance, and thus provide results that meet the user's needs. In application interaction, understanding user intent is crucial for enhancing user experience. By training intent recognition models with our constructed datasets, applications can better predict and respond to user actions. For example, in a smart assistant application, when a user issues a voice command to "set a reminder for a video conference at 9 am tomorrow," the intent recognition model can accurately understand the command and perform the corresponding operation, thereby improving the application's intelligence and user satisfaction.

To further process this dataset for other tasks or domains, we recommend using popular Python libraries, such as Pandas and NumPy.

Code availability

The code for replicating the experiment is available online via GitHub repository (https://github.com/lijie19740136878/BINS). The experiment's frontend collects users' voice inputs through a voice assistant system and transcribes them into text in real-time, which is then stored in a file. The code used for technical validation is developed in Python and is also available in the github repository mentioned above. If you need to use Capital One's DataProfiler, you can refer to the website (https://capitalone.github.io/DataProfiler/docs/0.9.0/html/profiler.html) or the GitHub online repository (https://github.com/capitalone/DataProfiler). Before using DataProfiler, you should install the corresponding Python package. For information on using the pyspellchecker, you can access it on GitHub (https://github.com/barrust/pyspellchecker).

Received: 5 November 2024; Accepted: 28 February 2025;
Published online: 11 March 2025

References

1. Jagatheesaperumal, S. K., Rahouti, M., Ahmad, K., Al-Fuqaha, A. & Guizani, M. The duo of artificial intelligence and big data for industry 4.0: Applications, techniques, challenges, and future research directions. *IEEE Internet Things J.* **9**, 12861–12885, https://doi.org/10.1109/JIOT.2021.3139827 (2022).
2. Gomes, P. H. *et al.* Intent-driven closed loops for autonomous networks. *J. ICT Standardization* **9**, 257–290 (2021). https://journals.riverpublishers.com/index.php/JICTS/article/view/5829.
3. Keimer, A. & Bayen, A. Routing on traffic networks incorporating past memory up to real-time information on the network state. *Annu. Rev. Control. Robotics, Auton. Syst.* **3**, 151–172, https://doi.org/10.1146/annurev-control-091319-125444 (2020).
4. Prag, K., Woolway, M. & Celik, T. Toward data-driven optimal control: A systematic review of the landscape. *IEEE Access* **10**, 32190–32212, https://doi.org/10.1109/ACCESS.2022.3160709 (2022).
5. Xu, S., Qian, Y. & Hu, R. Q. Data-driven network intelligence for anomaly detection. *IEEE Netw.* **33**, 88–95, https://doi.org/10.1109/MNET.2019.1800358 (2019).
6. Orlandi, B. *et al.* From a business intent to a network deployment with user-friendly interfaces and natural language processing. In *2024 27th Conference on Innovation in Clouds, Internet and Networks (ICIN)*, 91–93, (2024).
7. Sai, Z. *et al.* Intent-oriented network slicing with hypergraphs. IEEE Netw. 1–1, https://doi.org/10.1109/MNET.2024.3454117 (2024).
8. Wang, D., Su, R. & Zhang, S. An intent-based smart slicing framework for vertical industry in b5g networks. In *2021 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, 389–394, https://doi.org/10.1109/ICCCWorkshops52231.2021.9538888 (2021).
9. Mcnamara, J. *et al.* NLP powered intent-based network management for private 5G networks. *IEEE Access* **11**, 36642–36657, https://doi.org/10.1109/ACCESS.2023.3265894 (2023).
10. Chang, X. *et al.* KID: Knowledge graph-enabled intent-driven network with digital twin. In *2022 27th Asia Pacific Conference on Communications (APCC)*, 272–277, https://doi.org/10.1109/APCC55198.2022.9943779 (2022).
11. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding (2019). https://arxiv.org/abs/1810.04805.
12. Jacobs, A. S. *et al.* Hey, lumi! Using natural language for intent-based network management. In *USENIX Annual Technical Conference*. https://api.semanticscholar.org/CorpusID:236992633 (2021).
13. Andrade-Hoz, J., Wang, Q. & Alcaraz-Calero, J. M. Infrastructure-wide and intent-based networking dataset for 5g-and-beyond ai-driven autonomous networks. *Sensors* **24**, 783, https://doi.org/10.3390/s24030783 (2024).
14. Leivadeas, A. & Falkner, M. A survey on intent-based networking. *IEEE Commun. Surv. & Tutorials* **25**, 625–655, https://doi.org/10.1109/COMST.2022.3215919 (2023).
15. Peng, C., Xia, F., Naseriparsa, M. & Osborne, F. Knowledge graphs: Opportunities and challenges. *Artificial intelligence review* **56**, 13071–13102, https://doi.org/10.1007/s10462-023-10465-9 (2023).
16. Bing, W., Sai, Z., Minghui, L., Wei, N. & Xianbin, W. Explainable Application Intent for Zero-Touch Networking: An Incorporation of Hypergraph and Transformer. IEEE Transactions on Commun. https://doi.org/10.1109/TCOMM.2025.3529260 (2025).
17. Abbas, K. *et al.* AI-driven data analytics and intent-based networking for orchestration and control of b5g consumer electronics services. *IEEE Transactions on Consumer Electron.* **70**, 2155–2169, https://doi.org/10.1109/TCE.2023.3324010 (2024).
18. Zhang, G. *et al.* An intent-based routing scheme in satellite IoT. Springer Nat. Switz. 157–171, https://doi.org/10.1007/978-3-031-23141-4_12 (2022).
19. Singh, A., Aujla, G. S. & Bali, R. S. Intent-based network for data dissemination in software-defined vehicular edge computing. IEEE Transactions on Intell. *Transp. Syst.* **22**, 5310–5318, https://doi.org/10.1109/TITS.2020.3002349 (2021).
20. Njah, Y., Leivadeas, A., Violos, J. & Falkner, M. Toward intent-based network automation for smart environments: A healthcare 4.0 use case. *IEEE Access* **11**, 136565–136576, https://doi.org/10.1109/ACCESS.2023.3338189 (2023).
21. De Alwis, C. *et al.* Towards 6G: Key technological directions. *ICT Express* **9**, 525–533, https://doi.org/10.1016/j.icte.2022.10.005 (2023).
22. Wang, H. H. Speech recorder and translator using google cloud speech-to-text and translation. *J. IT Asia* **9**, 11–28, https://doi.org/10.33736/jita.2815.2021 (2021).
23. Jaberi, S., Atwood, J. W. & Paquet, J. ASSL as an intent expression language for autonomic intent-driven networking. In *2023 19th International Conference on Network and Service Management (CNSM)*, 1–5, https://doi.org/10.23919/CNSM59352.2023.10327860 (2023).
24. Li, C. *et al.* Intent Classification. RFC 9316, https://doi.org/10.17487/RFC9316 (2022).
25. Barrus, T. pyspellchecker: A pure python spell checker based on pydictionary. https://pypi.org/project/pyspellchecker/.
26. Yang, C. *et al.* Smart intent-driven network management. *IEEE Commun. Mag.* **61**, 106–112, https://doi.org/10.1109/MCOM.002.2200119 (2023).
27. Hu, J., Yang, W., Yang, H., Wei, S. & Sun, Z. Named entity recognition method for power equipment based on BERT-BiLSTM-CRF. In *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 1–6, https://doi.org/10.1109/DASC/PiCom/CBDCom/CyberSciTech53734.2022.00045 (2022).

28. Yao, X., Hao, X., Liu, R., Li, L. & Guo, X. Agcner, the first large-scale chinese named entity recognition dataset for agricultural diseases and pests. *Sci. Data* **11**, 769, https://doi.org/10.1038/s41597-024-03578-5 (2024).
29. Jie, L., Sai, Z., Yanglong, S., Hongfeng, G. & Wei, N. Business intent and network slicing correlation dataset from data-driven perspective. *Figshare* https://doi.org/10.6084/m9.figshare.27311538 (2025).
30. Blagec, K., Barbosa-Silva, A., Ott, S. & Samwald, M. A curated, ontology-based, large-scale knowledge graph of artificial intelligence tasks and benchmarks. *Sci. Data* **9**, 322, https://doi.org/10.1038/s41597-022-01435-x (2022).
31. Truong, A., Walters, A. & Goodsitt, J. Sensitive data detection with high-throughput neural network models for financial institutions. *CoRR* abs/2012.09597. https://arxiv.org/abs/2012.09597 (2020).
32. Li, J., Shi, Y. & Li, S. Analysis of beijing traffic violations based on the bert-crf model. *Promet - Traffic & Transportation* **36**, 279–293, https://doi.org/10.7307/ptt.v36i2.366 (2024).
33. Meng, F., Yang, S., Wang, J., Xia, L. & Liu, H. Creating knowledge graph of electric power equipment faults based on bert-bilstm-crf model. *Journal of electrical engineering & technology* **17**, 2507–2516, https://doi.org/10.1007/s42835-022-01032-3 (2022).
34. He, P., Liu, X., Gao, J. & Chen, W. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations*, https://openreview.net/pdf?id=sE7-XhLxHA (2023).
35. Yan, R., Jiang, X., Wang, W., Dang, D. & Su, Y. Materials information extraction via automatically generated corpus. *Sci. Data* **9**, 401, https://doi.org/10.1038/s41597-022-01492-2 (2022).
36. Katz, U., Vetzler, M., Cohen, A. & Goldberg, Y. NERetrieve: Dataset for next generation named entity recognition and retrieval. *Association for Computational Linguistics, Singapore*, 3340–3354, https://doi.org/10.18653/v1/2023.findings-emnlp.218 (2023).

Acknowledgements

This work is supported by the National Natural Science Foundation of China (62361011, 62462012, U24A20246).

Author contributions

J.L. and S.Z. designed the research and drafted the paper, Y.S. producted the figures and reviewed the manuscript, S.Z. and H.G. provided the research ideas, W.N reviewed the manuscript and analyzed the results.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to S.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.