FEDERATED KNOWLEDGE TRANSFER THROUGH KNOWLEDGE EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

The abstract paragraph should be indented 1/2 inch (3 picas) on both left and right-hand margins. Use 10 point type, with a vertical spacing of 11 points. The word ABSTRACT must be centered, in small caps, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.The abstract paragraph should be indented 1/2 inch (3 picas) on both left and right-hand margins. Use 10 point type, with a vertical spacing of 11 points. The word ABSTRACT must be centered, in small caps, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.The abstract paragraph should be indented 1/2 inch (3 picas) on both left and right-hand margins. Use 10 point type, with a vertical spacing of 11 points. The word ABSTRACT must be centered, in small caps, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 INTRODUCTION

Federated learning (FL) has emerged as a foundational paradigm for collaborative machine learning, enabling the training of powerful models on decentralized data without compromising user privacy (McMahan et al., 2017). The predominant approach, Federated Averaging (FedAvg), focuses on aggregating model parameters to build a single, unified global model (Kairouz et al., 2021). However, this parameter-averaging scheme suffers from well-known limitations, including a strict requirement for model homogeneity, high communication costs, and significant performance degradation in the presence of heterogeneous data distributions. To address or mitigate these weaknesses, the Federated Knowledge Transfer (FKT) subfield has emerged as a compelling alternative. FKT explores mechanisms for sharing higher-level insights and capabilities across clients, rather than raw model updates, though this may introduce a different set of trade-offs (Zhang et al., 2021; Mora et al., 2024).

Existing FKT methods, such as federated distillation and representation alignment, have demonstrated effectiveness in various scenarios (Li & Wang, 2019; Yu et al., 2025). These approaches primarily transfer coarse-grained knowledge, such as soft labels or model representations, between clients or between clients and the server (Hinton et al., 2015; Guo et al., 2024). While they have made significant contributions, they are inherently inefficient and costly, demanding multiple rounds of iterative training and high communication overhead (Kairouz et al., 2021; Almanifi et al., 2023). More critically, the lack of knowledge granularity makes it difficult to target and integrate specific information (Li et al., 2020). This can lead to negative transfer or even catastrophic forgetting, where new knowledge degrades previously learned critical information (McCloskey & Cohen, 1989; Wang et al., 2025b). There is a compelling need for a more precise, efficient, and targeted mechanism for knowledge transfer within the federated learning framework.

To overcome these limitations of inefficiency and imprecision, we turn to the rapidly advancing field of Knowledge Editing (KE) for large models. Knowledge Editing provides a set of powerful techniques to directly modify the behavior of a model to reflect new or corrected information, doing so with surgical precision and without the need for costly, full-scale retraining (Wang et al., 2025a; Meng et al., 2023). A particularly promising class of KE methods are memory-based approaches, which associate an external, nonparametric knowledge base with a frozen base model. New knowledge is stored as explicit key-value pairs in this external memory, and a lightweight retrieval mechanism is used to invoke this knowledge at inference time (Zhao et al., 2023; Mitchell et al., 2022). This approach offers an elegant solution to the challenges in FKT: it is highly efficient, as it involves no gradient-based training, and it is precise, as it isolates new knowledge in an external store, thereby preventing interference with the model's existing knowledge.

However, while KE provides a powerful mechanism for updating a model, it operates on a crucial assumption: the specific piece of knowledge to be added or modified, the (subject, relation, object) tuple, is known beforehand. In the context of federated knowledge transfer, this assumption no longer holds. The central, unaddressed challenge is to determine which pieces of knowledge from a distributed network of clients are the most valuable and worthy of being transferred and edited into another model's knowledge base.

In this paper, we bridge this critical gap by proposing a novel paradigm: **Federated Knowledge Transfer through Knowledge Editing (FKT-KE)**. Our framework reimagines FKT as a decentralized process of knowledge discovery, consolidation, and dynamic activation, inspired by cognitive memory models. Instead of transferring coarse-grained representations, clients broadcast lightweight predictions on a shared public dataset. Each client then captures all conflicting predictions from its peers, treating them as potential memories. These "memory traces" are consolidated into a local, non-parametric knowledge base and annotated with metrics reflecting their initial **Encoding Strength** (the peer's confidence) and **Semantic Resonance** (the degree of consensus across the network). At inference time, a dynamic activation mechanism selects the most potent knowledge, allowing for a highly efficient, precise, and context-aware transfer of individual knowledge units. This enables clients to learn from the collective expertise of the network without the drawbacks of traditional FL and FKT methods.

Our contributions are threefold:

- **A New Paradigm for Federated Knowledge Transfer via Knowledge Editing:** We introduce FKT-KE, a novel paradigm that is the first to apply Knowledge Editing techniques to knowledge transfer in federated learning. In this framework, each client maintains a private, local, non-parametric memory, and FKT is reframed as a decentralized process of knowledge discovery, consolidation, and editing. Collaboration occurs without parameter aggregation or gradient sharing; instead, clients broadcast lightweight predictions on a shared public dataset. This approach is designed to achieve a highly efficient, precise, and targeted transfer of individual knowledge units, enabling clients to learn from the collective expertise of the network.
- **A Cognitive-Inspired Mechanism for Dynamic Knowledge Selection:** To address the challenge of identifying and applying the most valuable knowledge in a distributed network, we propose a novel, two-stage mechanism inspired by human memory.
    1. **Broad Ingestion & Consolidation:** Instead of pre-filtering based on simple heuristics, each client initially ingests all knowledge that conflicts with its own predictions. These candidates are then consolidated into local "memory traces" through a continuous prompt learning process.
    2. **Dynamic Activation:** Each trace is annotated with its **Encoding Strength** (derived from the source peer's confidence) and its **Semantic Resonance** (derived from network consensus). At inference time, we introduce the concept of an **Activation Potential**, a score that dynamically weighs a memory's contextual similarity to the current query, its encoding strength, and its resonance. The client activates and applies the single memory trace with the highest potential. This decentralized, context-aware selection process ensures that only the most relevant and reliable knowledge is used, effectively mitigating negative transfer.
- **Comprehensive Evaluation and Analysis:** We conduct extensive experiments on multiple text classification datasets to demonstrate the superiority of our framework.

2 Related Work

Our work is positioned at the confluence of two dynamic research areas: Federated Knowledge Transfer and Knowledge Editing. We review the state-of-the-art in both domains, underscoring the specific challenges and opportunities that motivate our framework.

2.1 Federated Knowledge Transfer

Federated Knowledge Transfer (FKT) has arisen as a direct response to the inherent limitations of classical Federated Averaging (FedAvg) (Zhang et al., 2021). While FedAvg is foundational, its performance can degrade significantly under statistical heterogeneity (non-IID data), a common real-world scenario that causes client-level models to drift apart (Lu et al., 2024; Kairouz et al., 2021). FKT aims to surmount this challenge by shifting from aggregating model parameters to sharing higher-level knowledge representations (Zhang et al., 2021).

A prominent FKT paradigm is **Federated Distillation (FD)**, which leverages knowledge distillation (Hinton et al., 2015) to enable collaboration, even among clients with heterogeneous model architectures (Li & Wang, 2019; Lin et al., 2020). This approach has been actively developed, with recent methods focusing on generative techniques to create more diverse knowledge for distillation (Zhu et al., 2021) or improving the efficiency of the distillation process itself (Itahara et al., 2023). Another major thrust is **Federated Representation Learning (FRL)**, where clients collaborate by aligning their local feature spaces (Collins et al., 2021). This is often achieved by sharing and regularizing local representations or their prototypes (Tan et al., 2022), with recent studies exploring more advanced techniques like contrastive learning to further improve representation quality on non-IID data (Shi et al., 2023; Mu et al., 2023).

These FKT methodologies have undeniably advanced the field, providing robust solutions for model heterogeneity and non-IID data challenges. However, the knowledge they transfer—be it high-dimensional logit vectors or embedded feature prototypes—is inherently coarse-grained and holistic (Zhang et al., 2021). This lack of granularity makes the targeted transfer of discrete, factual information inefficient and can lead to knowledge dilution, where valuable, specific insights are obscured by irrelevant information from other clients (Wang et al., 2025b). Our work addresses this gap by proposing a mechanism for transferring fine-grained, individual knowledge units.

2.2 Knowledge Editing

Knowledge Editing (KE) has emerged as a vibrant field dedicated to efficiently and precisely modifying the knowledge within pre-trained models, serving as a low-cost alternative to full retraining (Zhang et al., 2024; Wang et al., 2025a). Research in KE has predominantly followed two major technical routes.

The first route involves **parameter-modifying methods**, which directly alter a model's weights to inject or erase information. This line of work has progressed from early approaches that applied localized updates (Mitchell et al., 2021) to a more sophisticated "locate-then-edit" methodology (Meng et al., 2022). This methodology, which pinpoints knowledge-storing neurons or layers (often in the Feed-Forward Networks) and applies a constrained update, is central to powerful editors like ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023). The latest research has pushed this boundary further, producing editors like PMET (Li et al., 2024) and EMMET (Gupta et al., 2024) capable of reliably scaling to thousands of edits. The second route focuses on **memory-augmented methods**, which preserve the integrity of the base model by storing new knowledge in an external, non-parametric memory (Zhao et al., 2023; Mitchell et al., 2022). Methods like SERAC (Mitchell et al., 2022) explicitly retrieve edited facts from a memory store at inference time, effectively preventing interference with the model's existing knowledge. A related concept uses in-context learning to provide ephemeral, demonstration-based knowledge without any weight modification (Qi et al., 2024).

From direct weight manipulation to external memory augmentation, the KE toolkit offers an impressive array of solutions for updating model knowledge. Yet, a crucial precondition underpins their entire operational model: the specific piece of knowledge to be edited is always supplied as an input (Zhang et al., 2024). This assumption does not hold in a federated learning context, where the objective is to first discover novel and valuable knowledge from a distributed network of clients. Our work confronts this fundamental challenge: we address the question of *what* knowledge to edit, a necessary precursor to leveraging the power of *how* to edit in a federated system.

3 THE FKT-KE FRAMEWORK

In this section, we present the detailed methodology of our proposed framework, Federated Knowledge Transfer through Knowledge Editing (FKT-KE). At its core, FKT-KE operates as a round-based, peer-to-peer process that enables efficient and precise knowledge dissemination without requiring any gradient-based training or model parameter aggregation. Each of the K clients in the network is equipped with its own task model M_k and a standardized set of pre-trained, frozen model editing modules. This standardized toolkit ensures that all clients can process, store, and apply external knowledge in a consistent manner. The entire process unfolds in four distinct phases within each knowledge transfer round, as detailed below.

3.1 PHASE 1: LOCAL INFERENCE AND PEER-TO-PEER BROADCAST

At the commencement of a round, all clients perform inference on a shared batch of public samples $\{x_i\}$ drawn from a public dataset $\mathcal{D}_{pub}$. For each sample x_i, every client k generates a prediction $y_{i,k}$ and a corresponding confidence score $c_{i,k}$, which is the softmax probability of the predicted class. Following local inference, each client broadcasts its list of prediction-confidence tuples, $\{(y_{i,k}, c_{i,k})\}$, to all other clients in the network. This broadcast of lightweight prediction metadata constitutes the sole communication step in our framework, thereby eliminating the costly exchange of model parameters and gradients typical of conventional federated learning algorithms.

3.2 PHASE 2: PRELIMINARY SCREENING AND STORAGE OF CONFLICTING KNOWLEDGE

Upon receiving the broadcasts from all peers, each client k independently evaluates the incoming information to preliminarily identify and store all knowledge it may currently lack. For each sample x_i, the client examines the prediction from every peer client j (where $j \neq k$).

A knowledge statement k_t, represented by the sample-label pair $(x_i, y_{i,j})$, is considered a candidate for ingestion and is stored in the local knowledge base $\mathcal{KB}_k$ if and only if its prediction conflicts with client k's own prediction. The ingestion rule is as follows:

$$\text{store}(k_t) \quad \text{if} \quad (y_{i,j} \neq y_{i,k})$$

This process ensures that all information conflicting with local judgment is captured as a copy, building a comprehensive candidate pool for the subsequent selection phase.

3.3 PHASE 3: KNOWLEDGE CONSOLIDATION AND MEMORY TRACE ANNOTATION

Once a candidate knowledge statement k_t is identified, it undergoes a process analogous to **memory consolidation**, where it is transformed and durably stored within the client's local knowledge base, $\mathcal{KB}_k$. This phase ensures that ephemeral information is encoded into a stable and retrievable format. The client utilizes its pre-trained editing modules to convert k_t into a structured key-value pair, or what we term a **"memory trace"**.

Knowledge Encoding (Key Generation): The process of creating a key is akin to semantic encoding in human memory. A knowledge statement k_t is passed through a frozen knowledge encoder f_{rm} (e.g., RoBERTa) and a key-generating MLP (MLP_K) to produce a dense vector representation, r_{k_t}. This operation is formally defined as:

$$r_{k_t} = \text{MLP}_K(f_{rm}(k_t)) \tag{1}$$

This vector r_{k_t} serves as the semantic address or "engram" of the memory trace, enabling efficient, content-addressable retrieval based on its meaning.

Continuous Prompt Transformation (Value Generation): The knowledge representation r_{k_t} is then transformed into a continuous prompt p_{k_t}, which functions as the procedural component of the memory trace—the "how-to" guide for the model. This is handled by a prompt transformer (MLP_P), generating a low-dimensional vector that can steer the LLM's behavior:

$$p_{k_t} = f_{\text{reshape}}(\text{MLP}_P(r_{k_t})) \tag{2}$$

This continuous prompt is the mechanism through which a declarative memory (the knowledge) is translated into an operational change in the model's output.

Memory Trace Annotation: Each newly consolidated memory trace (r_{k_t}, p_{k_t}) is annotated with two critical metadata attributes that determine its future accessibility:

- **Encoding Strength (Static):** This attribute is set to the confidence score $c_{i,j}$ from the source client j. In cognitive terms, this represents the initial salience or clarity of an experience. A memory trace encoded with high strength is considered more vivid and is prioritized during initial storage, reflecting a higher perceived likelihood of being correct.
- **Semantic Resonance (Static):** This attribute is an integer counter, $N_{i,j} = |\{m \mid y_{i,m} = y_{i,j}, m \in \{1, ..., K\}\}|$, representing the number of peers who independently asserted the same fact. This metric is analogous to the concept of social reinforcement in memory. A high resonance score indicates that the knowledge is coherent with the collective understanding of the network, strengthening its trace and increasing its probability of being recalled later.

3.4 PHASE 4: DYNAMIC MEMORY ACTIVATION AND ON-THE-FLY EDITING

During inference, when faced with a new input x_{new}, the client's model does not simply retrieve information; it performs a **dynamic memory activation** process to determine which, if any, stored memory trace is most relevant and potent for the current context.

Query Formulation and Activation Potential Calculation: The client first encodes the new input x_{new} into a query representation $\tilde{r}_q$ using a query encoder MLP_Q:

$$\tilde{r}_q = \text{MLP}_Q(f_{rm}(x_{new})) \tag{3}$$

Instead of a simple similarity search, the client calculates an **Activation Potential ($\mathcal{A}$)** for each memory trace k_j in its knowledge base $\mathcal{KB}_k$. This potential is a function of three factors: the contextual relevance of the memory, its initial encoding strength, and its semantic resonance within the network. The activation potential is calculated as follows:

$$\mathcal{A}(k_j, \tilde{r}_q) = \lambda_{\text{sim}} \cdot \text{sim}(\tilde{r}_q, r_{k_j}) + \lambda_E \cdot S_E(k_j) + \lambda_R \cdot \log(1 + S_R(k_j)) \tag{4}$$

where $\text{sim}(\cdot)$ is the cosine similarity, $S_E(k_j)$ is the Encoding Strength, $S_R(k_j)$ is the Semantic Resonance, and λ values are hyperparameters that balance the weight of each component. The memory trace k^* with the highest activation potential is selected for use:

$$k^* = \arg\max_{k_j \in \mathcal{KB}_k} \mathcal{A}(k_j, \tilde{r}_q) \tag{5}$$

Prompt Injection (Behavioral Realization): Once the memory trace with the highest activation potential, k^*, is identified, its procedural component—the optimal prompt $p^*_{k^*}$—is injected into the model's inference pathway. This is analogous to a retrieved memory influencing a cognitive response. The prompt is prepended to the input query's word embeddings:

$$\textbf{Input}_{\text{LLM}} = [p^*_{k^*} \oplus f_{\text{emb}}(x_{\text{new}})] \tag{6}$$

where f_{emb} is the model's embedding layer and $\oplus$ denotes concatenation. This on-the-fly ”recollection” allows the model to fluidly integrate expert knowledge from its peers, effectively ”editing” its behavior for a specific instance, much like how human memory shapes our real-time decisions without requiring fundamental changes to our brain's structure.

REFERENCES

Omair Rashed Abdulwareth Almanifi, Chee-Onn Chow, Mau-Luen Tham, Joon Huang Chuah, and Jeevan Kanesan. Communication and computation efficiency in Federated Learning: A survey. *Internet of Things*, 22:100742, 2023.

Liam Collins, Hamed Hassani, and Sewoong Oh. Exploiting Shared Representations for Personalized Federated Learning. In *International Conference on Machine Learning (ICML)*, 2021.

Wei Guo, Fuzhen Zhuang, Xiao Zhang, Yiqi Tong, and Jin Dong. A comprehensive survey of federated transfer learning: Challenges, methods and applications. *Frontiers of Computer Science*, 18(6):186356, 2024. doi: 10.1007/s11704-024-40065-x.

Akshat Gupta, Dev Sajnani, and Gopala Anumanchipalli. A Unified Framework for Model Editing. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings 2024*, pp. 15403–15418. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.903.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

Sohei Itahara, Takayuki Nishio, Yusuke Koda, Masahiro Morikura, and Koji Yamamoto. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Transactions on Mobile Computing*, 22(1):191–205, 2023.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawit, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14 (1–2):1–210, 2021. ISSN 1935-8237. doi: 10.1561/2200000083.

Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation, 2019.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, and Ameet Talwalkar. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, pp. 429–450, 2020.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. PMET: Precise Model Editing in a Transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17): 18564–18572, 2024. doi: 10.1609/aaai.v38i17.29818.

Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20. Curran Associates Inc., 2020. ISBN 9781713829546.

Zili Lu, Heng Pan, Yueyue Dai, Xueming Si, and Yan Zhang. Federated learning with non-iid data: A survey. *IEEE Internet of Things Journal*, 11(11):19188–19209, 2024. doi: 10.1109/JIOT.2024.3376548.

Michael McCloskey and Neal J. Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In Gordon H. Bower (ed.), *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Academic Press, 1989. doi: 10.1016/S0079-7421(08)60536-8.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In S. Koyejo (ed.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372. Curran Associates, Inc., 2022.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass editing memory in a transformer. *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast Model Editing at Scale. In *International Conference on Learning Representations*, 2021.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15817–15831, 17–23 Jul 2022.

Alessio Mora, Irene Tenison, Paolo Bellavista, and Irina Rish. Knowledge Distillation in Federated Learning: A Practical Guide. In *Thirty-Third International Joint Conference on Artificial Intelligence {IJCAI-24}*. International Joint Conferences on Artificial Intelligence Organization, 2024. doi: 10.24963/ijcai.2024/905.

Xutong Mu, Yulong Shen, Ke Cheng, Xueli Geng, Jiaxuan Fu, Tao Zhang, and Zhiwei Zhang. FedProc: Prototypical contrastive federated learning on non-IID data. *Future Generation Computer Systems*, 143:93–104, 2023. doi: 10.1016/j.future.2023.01.019.

Siyuan Qi, Bangcheng Yang, Kailin Jiang, Xiaobo Wang, Jiaqi Li, Yifan Zhong, Yaodong Yang, and Zilong Zheng. In-context editing: Learning knowledge from self-induced distributions. *CoRR*, abs/2406.11194, 2024. URL `https://doi.org/10.48550/arXiv.2406.11194`.

Xiaorong Shi, Liping Yi, Xiaoguang Liu, and Gang Wang. Ffedcl: Fair federated learning with contrastive learning. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10095453.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. FedProto: Federated Prototype Learning across Heterogeneous Clients. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8432–8440, 2022. doi: 10.1609/aaai.v36i8.20819.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge Editing for Large Language Models: A Survey. *ACM Computing Surveys*, 57(3):1–37, 2025a. doi: 10.1145/3698590.

Zhenyi Wang, Enneng Yang, Li Shen, and Heng Huang. A comprehensive survey of forgetting in deep learning beyond continual learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(3): 1464–1483, March 2025b. ISSN 0162-8828. doi: 10.1109/TPAMI.2024.3498346.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021. doi: 10.1016/j.knosys.2021.106775.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. A comprehensive study of knowledge editing for large language models, 2024. URL `https://arxiv.org/abs/2401.01286`.

Yao Zhao, Mikhail Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J Liu. Calibrating sequence likelihood improves conditional language generation. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12878–12889. PMLR, 18–24 Jul 2021.

A Appendix

You may include other additional sections here.