(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 December 2024 (19.12.2024)

WIPO | PCT

(10) International Publication Number
WO 2024/258705 A2

(51) International Patent Classification:
Not classified

(21) International Application Number:
PCT/US2024/032545

(22) International Filing Date:
05 June 2024 (05.06.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/521,122 15 June 2023 (15.06.2023) US

(71) Applicant: **VISA INTERNATIONAL SERVICE ASSOCIATION** [US/US]; P.O. Box 8999, San Francisco, California 94128 (US).

(72) Inventors: **CHEN, Huiyuan**; Visa International Service Association, P.O. Box 8999, San Francisco, California 94128 (US). **YEH, Michael**; Visa International Service Association, P.O. Box 8999, San Francisco, California 94128 (US). **LAI, Vivian, Wan Yin**; Visa International Service Association, P.O. Box 8999, San Francisco, California 94128 (US). **ZHENG, Yan**; Visa International Service Association, P.O. Box 8999, San Francisco, California 94128 (US). **DAS, Mahashweta**; Visa International Service Association, P.O. Box 8999, San Francisco, California 94128 (US). **YANG, Hao**; Visa International Service Association, P.O. Box 8999, San Francisco, California 94128 (US).

(74) Agent: **PREPELKA, Nathan, J.** et al.; The Webb Law Firm, One Gateway Center, 420 Ft. Duquesne Blvd., Suite 1200, Pittsburgh, Pennsylvania 15222 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM,

(54) Title: METHOD, SYSTEM, AND COMPUTER PROGRAM PRODUCT FOR OPERATING A LARGE SCALE GRAPH TRANSFORMER MACHINE LEARNING MODEL NETWORK ARCHITECTURE

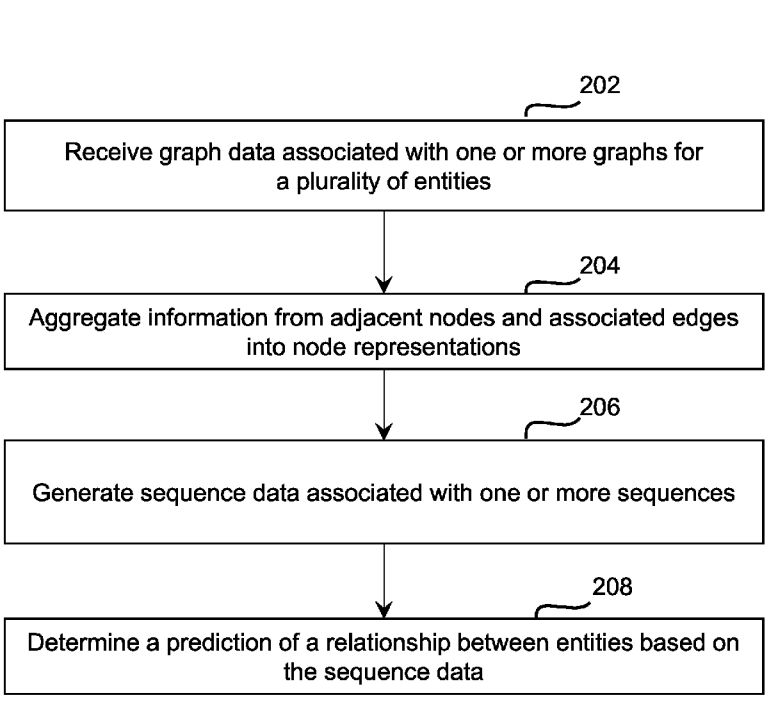

FIG. 2

(57) Abstract: Methods, systems, and computer program products are provided for operating a large scale graph transformer machine learning model network architecture. A method may include receiving graph data associated with a first graph for a first entity and graph data associated with a second graph for a second entity, aggregating information from nodes and edges of the first graph and information from nodes and edges of the second graph into a first plurality of node representations and a second plurality of node representations, generating first sequence data associated with a first sequence and second sequence data associated with a second sequence, providing the first sequence data associated with the first sequence data and the second sequence data as inputs to a transformer machine learning model, and determining a prediction of a relationship between the first entity and the second entity based on an output of the transformer

WO 2024/258705 A2

[Continued on next page]

DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

machine learning model.

METHOD, SYSTEM, AND COMPUTER PROGRAM PRODUCT FOR OPERATING A LARGE SCALE GRAPH TRANSFORMER MACHINE LEARNING MODEL NETWORK ARCHITECTURE

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims priority to United States Provisional Patent Application No. 63/521,122 filed on June 15, 2023, the disclosure of which is hereby incorporated by reference in its entirety.

BACKGROUND

1. Technical Field

[0002] The present disclosure relates generally to the use of transformer machine learning models and, in some non-limiting embodiments or aspects, to methods, systems, and computer program products for operating a large scale graph transformer machine learning model network architecture.

2. Technical Considerations

[0003] Neural networks (NNs) may be used for classification/prediction tasks in a variety of applications, such as facial recognition, fraud detection, disease diagnosis, navigation of self-driving cars, and/or the like. In such applications, NNs receive an input and generate predictions based on the input, for example, the identity of an individual, whether a payment transaction is fraudulent or not fraudulent, whether a disease is associated with one or more genetic markers, whether an object in a field of view of a self-driving car is in the self-driving car's path, and/or the like.

[0004] A form of NNs may include a graph neural network (GNN). A GNN may refer to an artificial neural network architecture for processing data that may be represented as graphs. A typical design element of a GNN is the use of pairwise message passing, such that graph node representations are iteratively updated based on information exchanged between neighboring graph nodes.

[0005] However, a GNN may have issues with regard to representation of a large-graph embedding. Such issues may include an over-smoothing, where a GNN may have a small number of graph convolutional layers, which may require an increase in the number of layers in the GNN. The increase in the number of layers of the GNN may cause embeddings of the GNN to converge to the same states and may require additional resources when performing inference related tasks. In addition, another issue may include over-squashing, where information flowing from distant nodes of a

graph may be a factor that limits the efficiency of message passing for tasks relying on long-distance interactions between the nodes of the graph.

SUMMARY

[0006] Accordingly, provided are improved methods, systems, and computer program products for operating a large scale graph transformer machine learning model network architecture.

[0007] According to non-limiting embodiments or aspects, provided is a computer implemented method for operating a large scale graph transformer machine learning model network architecture, that includes receiving, with at least one processor, graph data associated with a first graph for a first entity and graph data associated with a second graph for a second entity, wherein the first graph comprises a first plurality of nodes and edges, and wherein the second graph comprises a second plurality of nodes and edges; aggregating, with at least one processor, information from adjacent nodes and associated edges of the first graph into a first plurality of node representations; aggregating, with at least one processor, information from adjacent nodes and associated edges of the second graph into a second plurality of node representations; generating, with at least one processor, first sequence data associated with a first sequence of the first plurality of node representations; generating, with at least one processor, second sequence data associated with a second sequence of the second plurality of node representations; providing, with at least one processor, the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs to a transformer machine learning model; and determining, with at least one processor, a prediction of a relationship between the first entity and the second entity based on an output of the transformer machine learning model.

[0008] In some non-limiting embodiments or aspects, the method further includes generating the output of the transformer machine learning model based on the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs, wherein the output includes a first final embedding associated with the first entity, and a second final embedding associated with the second entity.

[0009] In some non-limiting embodiments or aspects, the method further includes calculating an inner product of the first final embedding associated with the first entity

and the second final embedding associated with the second entity, wherein determining the prediction of a relationship between the first entity and the second entity includes determining the prediction of a relationship between the first entity and the second entity based on the inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity.

[0010] In some non-limiting embodiments or aspects, the method further includes generating a first graph neural network based on the graph data associated with the first graph for the first entity; and generating a second graph neural network based on the graph data associated with the second graph for the second entity.

[0011] In some non-limiting embodiments or aspects, aggregating the information from adjacent nodes and associated edges of the first graph into the first plurality of node representations includes concatenating a first node embedding of the first graph neural network and a first positional embedding of the first graph neural network to provide the first plurality of node representations, and wherein aggregating the information from adjacent nodes and associated edges of the second graph into the second plurality of node representations includes concatenating a second node embedding of the second graph neural network and a second positional embedding of the second graph neural network to provide the second plurality of node representations.

[0012] In some non-limiting embodiments or aspects, the method further includes performing a first breadth first search of the first graph for the first entity to obtain the graph data associated with the first graph and performing a second breadth first search of the second graph for the second entity to obtain the graph data associated with the second graph.

[0013] In some non-limiting embodiments or aspects, the method further includes generating a recommendation for the first entity based on the prediction of a relationship between the first entity and the second entity and transmitting the recommendation to a user device of the first entity.

[0014] According to non-limiting embodiments or aspects, provided is a system for operating a large scale graph transformer machine learning model network architecture, that includes at least one processor configured to receive graph data associated with a first graph for a first entity and graph data associated with a second graph for a second entity, wherein the first graph comprises a first plurality of nodes and edges, and wherein the second graph comprises a second plurality of nodes and

edges; aggregate information from adjacent nodes and associated edges of the first graph into a first plurality of node representations; aggregate information from adjacent nodes and associated edges of the second graph into a second plurality of node representations; generate first sequence data associated with a first sequence of the first plurality of node representations; generate second sequence data associated with a second sequence of the second plurality of node representations; provide the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs to a transformer machine learning model; and determine a prediction of a relationship between the first entity and the second entity based on an output of the transformer machine learning model.

[0015] In some non-limiting embodiments or aspects, the at least one processor is further configured to generate the output of the transformer machine learning model based on the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs, wherein the output comprises: a first final embedding associated with the first entity, and a second final embedding associated with the second entity.

[0016] In some non-limiting embodiments or aspects, the at least one processor is further configured to calculate an inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity, wherein, when determining the prediction of a relationship between the first entity and the second entity, the at least one processor is configured to determine the prediction of a relationship between the first entity and the second entity based on the inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity.

[0017] In some non-limiting embodiments or aspects, the at least one processor is further configured to generate a first graph neural network based on the graph data associated with the first graph for the first entity; and generate a second graph neural network based on the graph data associated with the second graph for the second entity.

[0018] In some non-limiting embodiments or aspects, when aggregating the information from adjacent nodes and associated edges of the first graph into the first plurality of node representations, the at least one processor is configured to concatenate a first node embedding of the first graph neural network and a first positional embedding of the first graph neural network to provide the first plurality of

node representations, and wherein, when aggregating the information from adjacent nodes and associated edges of the second graph into the second plurality of node representations, the at least one processor is configured to concatenate a second node embedding of the second graph neural network and a second positional embedding of the second graph neural network to provide the second plurality of node representations.

[0019] In some non-limiting embodiments or aspects, the at least one processor is further configured to perform a first breadth first search of the first graph for the first entity to obtain the graph data associated with the first graph; and perform a second breadth first search of the second graph for the second entity to obtain the graph data associated with the second graph.

[0020] In some non-limiting embodiments or aspects, the at least one processor is further configured to generate a recommendation for the first entity based on the prediction of a relationship between the first entity and the second entity; and transmit the recommendation to the first entity.

[0021] According to non-limiting embodiments or aspects, provided is a computer program product for operating a large scale graph transformer machine learning model network architecture, that includes at least one non-transitory computer-readable medium including program instructions that, when executed by at least one processor, cause the at least one processor to receive graph data associated with a first graph for a first entity and graph data associated with a second graph for a second entity, wherein the first graph comprises a first plurality of nodes and edges, and wherein the second graph comprises a second plurality of nodes and edges; aggregate information from adjacent nodes and associated edges of the first graph into a first plurality of node representations; aggregate information from adjacent nodes and associated edges of the second graph into a second plurality of node representations; generate first sequence data associated with a first sequence of the first plurality of node representations; generate second sequence data associated with a second sequence of the second plurality of node representations; provide the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs to a transformer machine learning model; and determine a prediction of a relationship between the first entity and the second entity based on an output of the transformer machine learning model.

[0022] In some non-limiting embodiments or aspects, the program instructions further cause the at least one processor to generate the output of the transformer machine learning model based on the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs, wherein the output comprises: a first final embedding associated with the first entity, and a second final embedding associated with the second entity.

[0023] In some non-limiting embodiments or aspects, the program instructions further cause the at least one processor to calculate an inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity, wherein the one or more program instructions that cause the at least one processor to determine the prediction of a relationship between the first entity and the second entity, cause the at least one processor to determine the prediction of a relationship between the first entity and the second entity based on the inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity.

[0024] In some non-limiting embodiments or aspects, the program instructions further cause the at least one processor to generate a first graph neural network based on the graph data associated with the first graph for the first entity; and generate a second graph neural network based on the graph data associated with the second graph for the second entity.

[0025] In some non-limiting embodiments or aspects, the one or more program instructions that cause the at least one processor to aggregate the information from adjacent nodes and associated edges of the first graph into the first plurality of node representations, cause the at least one processor to concatenate a first node embedding of the first graph neural network and a first positional embedding of the first graph neural network to provide the first plurality of node representations, and wherein the one or more program instructions that cause the at least one processor to aggregate the information from adjacent nodes and associated edges of the second graph into the second plurality of node representations, cause the at least one processor to concatenate a second node embedding of the second graph neural network and a second positional embedding of the second graph neural network to provide the second plurality of node representations.

[0026] In some non-limiting embodiments or aspects, the program instructions further cause the at least one processor to perform a first breadth first search of the

first graph for the first entity to obtain the graph data associated with the first graph and perform a second breadth first search of the second graph for the second entity to obtain the graph data associated with the second graph.

[0027] In some non-limiting embodiments or aspects, the program instructions further cause the at least one processor to generate a recommendation for the first entity based on the prediction of a relationship between the first entity and the second entity; and transmit the recommendation to the first entity.

[0028] Further non-limiting embodiments or aspects will be set forth in the following numbered clauses:

[0029] Clause 1: A computer-implemented method, comprising: receiving, with at least one processor, graph data associated with a first graph for a first entity and graph data associated with a second graph for a second entity, wherein the first graph comprises a first plurality of nodes and edges, and wherein the second graph comprises a second plurality of nodes and edges; aggregating, with at least one processor, information from adjacent nodes and associated edges of the first graph into a first plurality of node representations; aggregating, with at least one processor, information from adjacent nodes and associated edges of the second graph into a second plurality of node representations; generating, with at least one processor, first sequence data associated with a first sequence of the first plurality of node representations; generating, with at least one processor, second sequence data associated with a second sequence of the second plurality of node representations; providing, with at least one processor, the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs to a transformer machine learning model; and determining, with at least one processor, a prediction of a relationship between the first entity and the second entity based on an output of the transformer machine learning model.

[0030] Clause 2: The computer-implemented method of clause 1, further comprising: generating the output of the transformer machine learning model based on the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs, wherein the output comprises: a first final embedding associated with the first entity, and a second final embedding associated with the second entity.

[0031] Clause 3: The computer-implemented method of clause 1 or 2, further comprising: calculating an inner product of the first final embedding associated with

the first entity and the second final embedding associated with the second entity, wherein determining the prediction of a relationship between the first entity and the second entity comprises: determining the prediction of a relationship between the first entity and the second entity based on the inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity.

[0032] Clause 4: The computer-implemented method of any of clauses 1-3, further comprising: generating a first graph neural network based on the graph data associated with the first graph for the first entity; and generating a second graph neural network based on the graph data associated with the second graph for the second entity.

[0033] Clause 5: The computer-implemented method of any of clauses 1-4, wherein aggregating the information from adjacent nodes and associated edges of the first graph into the first plurality of node representations comprises: concatenating a first node embedding of the first graph neural network and a first positional embedding of the first graph neural network to provide the first plurality of node representations, and wherein aggregating the information from adjacent nodes and associated edges of the second graph into the second plurality of node representations comprises: concatenating a second node embedding of the second graph neural network and a second positional embedding of the second graph neural network to provide the second plurality of node representations.

[0034] Clause 6: The computer-implemented method of any of clauses 1-5, further comprising: performing a first breadth first search of the first graph for the first entity to obtain the graph data associated with the first graph; and performing a second breadth first search of the second graph for the second entity to obtain the graph data associated with the second graph.

[0035] Clause 7: The computer-implemented method of any of clauses 1-6, further comprising: generating a recommendation for the first entity based on the prediction of a relationship between the first entity and the second entity; and transmitting the recommendation to a user device of the first entity.

[0036] Clause 8: A system, comprising: at least one processor configured to: receive graph data associated with a first graph for a first entity and graph data associated with a second graph for a second entity, wherein the first graph comprises a first plurality of nodes and edges, and wherein the second graph comprises a second

plurality of nodes and edges; aggregate information from adjacent nodes and associated edges of the first graph into a first plurality of node representations; aggregate information from adjacent nodes and associated edges of the second graph into a second plurality of node representations; generate first sequence data associated with a first sequence of the first plurality of node representations; generate second sequence data associated with a second sequence of the second plurality of node representations; provide the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs to a transformer machine learning model; and determine a prediction of a relationship between the first entity and the second entity based on an output of the transformer machine learning model.

[0037] Clause 9: The system of clause 8, wherein the at least one processor is further configured to: generate the output of the transformer machine learning model based on the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs, wherein the output comprises: a first final embedding associated with the first entity, and a second final embedding associated with the second entity.

[0038] Clause 10: The system of clause 8 or 9, wherein the at least one processor is further configured to: calculate an inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity, wherein, when determining the prediction of a relationship between the first entity and the second entity, the at least one processor is configured to: determine the prediction of a relationship between the first entity and the second entity based on the inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity.

[0039] Clause 11: The system of any of clauses 8-10, wherein the at least one processor is further configured to: generate a first graph neural network based on the graph data associated with the first graph for the first entity; and generate a second graph neural network based on the graph data associated with the second graph for the second entity.

[0040] Clause 12: The system of any of clauses 8-11, wherein, when aggregating the information from adjacent nodes and associated edges of the first graph into the first plurality of node representations, the at least one processor is configured to: concatenate a first node embedding of the first graph neural network and a first

positional embedding of the first graph neural network to provide the first plurality of node representations, and wherein, when aggregating the information from adjacent nodes and associated edges of the second graph into the second plurality of node representations, the at least one processor is configured to: concatenate a second node embedding of the second graph neural network and a second positional embedding of the second graph neural network to provide the second plurality of node representations.

[0041] Clause 13: The system of any of clauses 8-12, wherein the at least one processor is further configured to: perform a first breadth first search of the first graph for the first entity to obtain the graph data associated with the first graph; and perform a second breadth first search of the second graph for the second entity to obtain the graph data associated with the second graph.

[0042] Clause 14: The system of any of clauses 8-13, wherein the at least one processor is further configured to: generate a recommendation for the first entity based on the prediction of a relationship between the first entity and the second entity; and transmit the recommendation to the first entity.

[0043] Clause 15: A computer program product comprising at least one non-transitory computer-readable medium including program instructions that, when executed by at least one processor, cause the at least one processor to: receive graph data associated with a first graph for a first entity and graph data associated with a second graph for a second entity, wherein the first graph comprises a first plurality of nodes and edges, and wherein the second graph comprises a second plurality of nodes and edges; aggregate information from adjacent nodes and associated edges of the first graph into a first plurality of node representations; aggregate information from adjacent nodes and associated edges of the second graph into a second plurality of node representations; generate first sequence data associated with a first sequence of the first plurality of node representations; generate second sequence data associated with a second sequence of the second plurality of node representations; provide the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs to a transformer machine learning model; and determine a prediction of a relationship between the first entity and the second entity based on an output of the transformer machine learning model.

[0044] Clause 16: The computer program product of clause 15, wherein the program instructions further cause the at least one processor to: generate the output of the transformer machine learning model based on the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs, wherein the output comprises: a first final embedding associated with the first entity, and a second final embedding associated with the second entity.

[0045] Clause 17: The computer program product of clause 15 or 16, wherein the program instructions further cause the at least one processor to: calculate an inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity, wherein, when determining the prediction of a relationship between the first entity and the second entity, the at least one processor is configured to: determine the prediction of a relationship between the first entity and the second entity based on the inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity.

[0046] Clause 18: The computer program product of any of clauses 15-17, wherein the program instructions further cause the at least one processor to: generate a first graph neural network based on the graph data associated with the first graph for the first entity; and generate a second graph neural network based on the graph data associated with the second graph for the second entity.

[0047] Clause 19: The computer program product of any of clauses 15-18, wherein, when aggregating the information from adjacent nodes and associated edges of the first graph into the first plurality of node representations, the at least one processor is configured to: concatenate a first node embedding of the first graph neural network and a first positional embedding of the first graph neural network to provide the first plurality of node representations, and wherein, when aggregating the information from adjacent nodes and associated edges of the second graph into the second plurality of node representations, the at least one processor is configured to: concatenate a second node embedding of the second graph neural network and a second positional embedding of the second graph neural network to provide the second plurality of node representations.

[0048] Clause 20: The computer program product of any of clauses 15-19, wherein the program instructions further cause the at least one processor to: perform a first

breadth first search of the first graph for the first entity to obtain the graph data associated with the first graph; and perform a second breadth first search of the second graph for the second entity to obtain the graph data associated with the second graph.

[0049] Clause 21: The computer program product of any of clauses 15-20, wherein the program instructions further cause the at least one processor to: generate a recommendation for the first entity based on the prediction of a relationship between the first entity and the second entity; and transmit the recommendation to the first entity.

[0050] These and other features and characteristics of the present disclosure, as well as the methods of operation and functions of the related elements of structures and the combination of parts and economies of manufacture, will become more apparent upon consideration of the following description and the appended claims with reference to the accompanying drawings, all of which form a part of this specification, wherein like reference numerals designate corresponding parts in the various figures. It is to be expressly understood, however, that the drawings are for the purpose of illustration and description only and are not intended as a definition of the limits of the disclosed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0051] Additional advantages and details of the present disclosure are explained in greater detail below with reference to the exemplary embodiments that are illustrated in the accompanying figures, in which:

[0052] FIG. 1 is a diagram of a non-limiting embodiment or aspect of an environment in which systems, devices, products, apparatus, and/or methods, described herein, may be implemented, according to the principles of the present disclosure;

[0053] FIG. 2 is a flowchart of a non-limiting embodiment or aspect of a process for operating a large scale graph transformer machine learning model;

[0054] FIGS. 3A-3E are schematic diagrams of an exemplary implementation of a system and/or method for operating a large scale graph transformer machine learning model network architecture, according to some non-limiting embodiments or aspects;

[0055] FIG. 4 is a diagram of an exemplary environment in which systems, methods, and/or computer program products, described herein, may be implemented, according to some non-limiting embodiments or aspects; and

[0056] FIG. 5 is a schematic diagram of example components of one or more devices of FIG. 1 and/or FIG. 4, according to some non-limiting embodiments or aspects.

DETAILED DESCRIPTION

[0057] For purposes of the description hereinafter, the terms "end," "upper," "lower," "right," "left," "vertical," "horizontal," "top," "bottom," "lateral," "longitudinal," and derivatives thereof shall relate to the embodiments as they are oriented in the drawing figures. However, it is to be understood that the embodiments may assume various alternative variations and step sequences, except where expressly specified to the contrary. It is also to be understood that the specific devices and processes illustrated in the attached drawings, and described in the following specification, are simply exemplary embodiments or aspects of the disclosed subject matter. Hence, specific dimensions and other physical characteristics related to the embodiments or aspects disclosed herein are not to be considered as limiting.

[0058] Some non-limiting embodiments or aspects may be described herein in connection with thresholds. As used herein, satisfying a threshold may refer to a value being greater than the threshold, more than the threshold, higher than the threshold, greater than or equal to the threshold, less than the threshold, fewer than the threshold, lower than the threshold, less than or equal to the threshold, equal to the threshold, etc.

[0059] No aspect, component, element, structure, act, step, function, instruction, and/or the like used herein should be construed as critical or essential unless explicitly described as such. Also, as used herein, the articles "a" and "an" are intended to include one or more items and may be used interchangeably with "one or more" and "at least one." Furthermore, as used herein, the term "set" is intended to include one or more items (e.g., related items, unrelated items, a combination of related and unrelated items, and/or the like) and may be used interchangeably with "one or more" or "at least one." Where only one item is intended, the term "one" or similar language is used. Also, as used herein, the terms "has," "have," "having," or the like are intended to be open-ended terms. Further, the phrase "based on" is intended to mean "based

at least partially on" unless explicitly stated otherwise. In addition, reference to an action being "based on" a condition may refer to the action being "in response to" the condition. For example, the phrases "based on" and "in response to" may, in some non-limiting embodiments or aspects, refer to a condition for automatically triggering an action (e.g., a specific operation of an electronic device, such as a computing device, a processor, and/or the like).

[0060] As used herein, the term "acquirer institution" may refer to an entity licensed and/or approved by a transaction service provider to originate transactions (e.g., payment transactions) using a payment device associated with the transaction service provider. The transactions the acquirer institution may originate may include payment transactions (e.g., purchases, original credit transactions (OCTs), account funding transactions (AFTs), and/or the like). In some non-limiting embodiments or aspects, an acquirer institution may be a financial institution, such as a bank. As used herein, the term "acquirer system" may refer to one or more computing devices operated by or on behalf of an acquirer institution, such as a server computer executing one or more software applications.

[0061] As used herein, the term "account identifier" may include one or more primary account numbers (PANs), tokens, or other identifiers associated with a customer account. The term "token" may refer to an identifier that is used as a substitute or replacement identifier for an original account identifier, such as a PAN. Account identifiers may be alphanumeric or any combination of characters and/or symbols. Tokens may be associated with a PAN or other original account identifier in one or more data structures (e.g., one or more databases, and/or the like) such that they may be used to conduct a transaction without directly using the original account identifier. In some examples, an original account identifier, such as a PAN, may be associated with a plurality of tokens for different individuals or purposes.

[0062] As used herein, the term "communication" may refer to the reception, receipt, transmission, transfer, provision, and/or the like of data (e.g., information, signals, messages, instructions, commands, and/or the like). For one unit (e.g., a device, a system, a component of a device or system, combinations thereof, and/or the like) to be in communication with another unit means that the one unit is able to directly or indirectly receive information from and/or transmit information to the other unit. This may refer to a direct or indirect connection (e.g., a direct communication connection, an indirect communication connection, and/or the like) that is wired and/or

wireless in nature. Additionally, two units may be in communication with each other even though the information transmitted may be modified, processed, relayed, and/or routed between the first and second units. For example, a first unit may be in communication with a second unit even though the first unit passively receives information and does not actively transmit information to the second unit. As another example, a first unit may be in communication with a second unit if at least one intermediary unit processes information received from the first unit and communicates the processed information to the second unit. In some non-limiting embodiments or aspects, a message may refer to a network packet (e.g., a data packet and/or the like) that includes data. It will be appreciated that numerous other arrangements are possible.

[0063] As used herein, the term "computing device" may refer to one or more electronic devices configured to process data. A computing device may, in some examples, include the necessary components to receive, process, and output data, such as a processor, a display, a memory, an input device, a network interface, and/or the like. A computing device may be a mobile device. As an example, a mobile device may include a cellular phone (e.g., a smartphone or standard cellular phone), a portable computer, a wearable device (e.g., watches, glasses, lenses, clothing, and/or the like), a personal digital assistant (PDA), and/or other like devices. A computing device may also be a desktop computer or other form of non-mobile computer.

[0064] As used herein, the term "server" may refer to or include one or more computing devices that are operated by or facilitate communication and processing for multiple parties in a network environment, such as the Internet, although it will be appreciated that communication may be facilitated over one or more public or private network environments and that various other arrangements are possible. Further, multiple computing devices (e.g., servers, point-of-sale (POS) devices, mobile devices, etc.) directly or indirectly communicating in the network environment may constitute a "system."

[0065] As used herein, the term "system" may refer to one or more computing devices or combinations of computing devices (e.g., processors, servers, client devices, software applications, components of such, and/or the like). Reference to "a device," "a server," "a processor," and/or the like, as used herein, may refer to a previously-recited device, server, or processor that is recited as performing a previous step or function, a different device, server, or processor, and/or a combination of

devices, servers, and/or processors. For example, as used in the specification and the claims, a first device, a first server, or a first processor that is recited as performing a first step or a first function may refer to the same or different device, server, or processor recited as performing a second step or a second function.

[0066] As used herein, the term "issuer institution" may refer to one or more entities, such as a bank, that provide accounts to customers for conducting transactions (e.g., payment transactions), such as initiating credit and/or debit payments. For example, an issuer institution may provide an account identifier, such as a PAN, to a customer that uniquely identifies one or more accounts associated with that customer. The account identifier may be embodied on a portable financial device, such as a physical financial instrument, e.g., a payment card, and/or may be electronic and used for electronic payments. The term "issuer system" refers to one or more computer devices operated by or on behalf of an issuer institution, such as a server computer executing one or more software applications. For example, an issuer system may include one or more authorization servers for authorizing a transaction.

[0067] As used herein, the term "merchant" may refer to an individual or entity that provides goods and/or services, or access to goods and/or services, to customers based on a transaction, such as a payment transaction. The term "merchant" or "merchant system" may also refer to one or more computer systems operated by or on behalf of a merchant, such as a server computer executing one or more software applications.

[0068] As used herein, the term "payment device" may refer to an electronic payment device, a portable financial device (e.g., a payment card, such as a credit or debit card), a gift card, a smartcard, smart media, a payroll card, a healthcare card, a wristband, a machine-readable medium containing account information, a keychain device or fob, a radio frequency identification (RFID) transponder, a retailer discount or loyalty card, a cellular phone, an electronic wallet mobile application, a PDA, a pager, a security card, a computing device, an access card, a wireless terminal, a transponder, and/or the like. In some non-limiting embodiments or aspects, the payment device may include volatile or non-volatile memory to store information (e.g., an account identifier, a name of the account holder, and/or the like).

[0069] As used herein, a "point-of-sale (POS) device" may refer to one or more devices, which may be used by a merchant to conduct a transaction (e.g., a payment transaction) and/or process a transaction. For example, a POS device may include

one or more client devices. Additionally or alternatively, a POS device may include peripheral devices, card readers, scanning devices (e.g., code scanners), Bluetooth® communication receivers, near-field communication (NFC) receivers, RFID receivers, and/or other contactless transceivers or receivers, contact-based receivers, payment terminals, and/or the like. As used herein, a "point-of-sale (POS) system" may refer to one or more client devices and/or peripheral devices used by a merchant to conduct a transaction. For example, a POS system may include one or more POS devices and/or other like devices that may be used to conduct a payment transaction. In some non-limiting embodiments or aspects, a POS system (e.g., a merchant POS system) may include one or more server computers configured to process online payment transactions through webpages, mobile applications, and/or the like.

[0070] As used herein, the term "transaction service provider" may refer to an entity that receives transaction authorization requests from merchants or other entities and provides guarantees of payment, in some cases through an agreement between the transaction service provider and an issuer institution. For example, a transaction service provider may include a payment network such as Visa® or any other entity that processes transactions. The term "transaction processing system" may refer to one or more computer systems operated by or on behalf of a transaction service provider, such as a transaction processing server executing one or more software applications. A transaction processing server may include one or more processors and, in some non-limiting embodiments or aspects, may be operated by or on behalf of a transaction service provider.

[0071] Non-limiting embodiments or aspects of the present disclosure are directed to systems, methods, and computer program products for operating a large scale graph transformer machine learning model network architecture. In some non-limiting embodiments or aspects, a model management system may include at least one processor configured to receive graph data associated with a first graph for a first entity and graph data associated with a second graph for a second entity, where the first graph includes a first plurality of nodes and edges and the second graph includes a second plurality of nodes and edges, aggregate information from adjacent nodes and associated edges of the first graph into a first plurality of node representations, aggregate information from adjacent nodes and associated edges of the second graph into a second plurality of node representations, generate first sequence data associated with a first sequence of the first plurality of node representations, generate

second sequence data associated with a second sequence of the second plurality of node representations, provide the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs to a transformer machine learning model, and determine a prediction of a relationship between the first entity and the second entity based on an output of the transformer machine learning model (e.g., a machine learning model having a large scale graph transformer machine learning model network architecture). In some non-limiting embodiments or aspects, the at least one processor is further configured to perform a first breadth first search of the first graph for the first entity to obtain the graph data associated with the first graph and perform a second breadth first search of the second graph for the second entity to obtain the graph data associated with the second graph.

[0072] In some non-limiting embodiments or aspects, the at least one processor is further configured to generate the output of the transformer machine learning model based on the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs, where the output includes a first final embedding associated with the first entity and a second final embedding associated with the second entity. In some non-limiting embodiments or aspects, the at least one processor is further configured to calculate an inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity. In some non-limiting embodiments or aspects, when determining the prediction of a relationship between the first entity and the second entity, the at least one processor is configured to determine the prediction of a relationship between the first entity and the second entity based on the inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity.

[0073] In some non-limiting embodiments or aspects, the at least one processor is further configured to generate a first graph neural network based on the graph data associated with the first graph for the first entity and generate a second graph neural network based on the graph data associated with the second graph for the second entity. In some non-limiting embodiments or aspects, when aggregating the information from adjacent nodes and associated edges of the first graph into the first plurality of node representations, the at least one processor is configured to concatenate a first node embedding of the first graph neural network and a first

positional embedding of the first graph neural network to provide the first plurality of node representations. In some non-limiting embodiments or aspects, when aggregating the information from adjacent nodes and associated edges of the second graph into the second plurality of node representations, the at least one processor is configured to concatenate a second node embedding of the second graph neural network and a second positional embedding of the second graph neural network to provide the second plurality of node representations.

[0074] In this way, the model management system may provide for a large scale graph transformer machine learning model network architecture that solves issues relating to over-smoothing and/or over-squashing that may be present with graph neural network (GNNs). For example, the large scale graph transformer machine learning model network architecture may reduce the number of layers necessary to achieve accurate performance and may reduce the effect in a GNN that causes embeddings to converge to the same states. In addition, the large scale graph transformer machine learning model network architecture may allow for accurate and efficient information transfer between distant nodes of a graph, such that long-distance interactions between the nodes of the graph are accurately represented. With this, the model management system may generate more accurate recommendations with regard to an entity as compared to a typical GNN architecture.

[0075] For the purpose of illustration, in the following description, while the presently disclosed subject matter is described with respect to methods, systems, and computer program products for a large scale graph transformer machine learning model network architecture, which may be used in association with providing recommendations, one skilled in the art will recognize that the disclosed subject matter is not limited to the non-limiting embodiments or aspects disclosed herein. For example, the methods, systems, and computer program products described herein may be used with a wide variety of settings and/or for making determinations (e.g., predictions, classifications, regressions, and/or the like), such as for fraud detection/prevention, authorization, authentication, identification, feature selection, payment processing, and/or the like.

[0076] Referring now to FIG. 1, FIG. 1 is a diagram of example system 100 in which devices, systems, and/or methods, described herein, may be implemented. As shown in FIG. 1, system 100 includes model management system 102, machine learning (ML) model management database 104, user device 106, and communication network

108. Model management system 102, ML model management database 104, and/or user device 106 may interconnect (e.g., establish a connection to communicate) via wired connections, wireless connections, or a combination of wired and wireless connections.

[0077] Model management system 102 may include one or more devices capable of receiving information from and/or communicating information (e.g., directly via wired or wireless communication connection, indirectly via communication network 108, and/or the like) to ML model management database 104 and/or user device 106 via communication network 108. For example, model management system 102 may include a server, a group of servers, a cloud platform, and/or other like devices. In some non-limiting embodiments or aspects, model management system 102 may be associated with a transaction service provider system. For example, model management system 102 may be operated by a transaction service provider system. In another example, model management system 102 may be a component of user device 106. In another example, model management system 102 may include ML model management database 104. In some non-limiting embodiments or aspects, model management system 102 may be in communication with a data storage device (e.g., ML model management database 104), which may be local or remote to model management system 102. In some non-limiting embodiments or aspects, model management system 102 may be capable of receiving information from, storing information in, transmitting information to, and/or searching information stored in the data storage device.

[0078] ML model management database 104 may include one or more devices capable of receiving information from and/or communicating information (e.g., directly via wired or wireless communication connection, indirectly via communication network 108, and/or the like) to model management system 102 and/or user device 106. For example, ML model management database 104 may include a server, a group of servers, a desktop computer, a portable computer, a mobile device, and/or other like devices. In some non-limiting embodiments or aspects, ML model management database 104 may include a data storage device. In some non-limiting embodiments or aspects, ML model management database 104 may be capable of receiving information from, storing information in, communicating information to, or searching information stored in the data storage device. In some non-limiting embodiments or

aspects, ML model management database 104 may be part of model management system 102 and/or part of the same system as model management system 102.

[0079] User device 106 may include one or more devices capable of receiving information from and/or communicating information (e.g., directly via wired or wireless communication connection, indirectly via communication network 108, and/or the like) to model management system 102 and/or ML model management database 104. For example, user device 106 may include a computing device, such as a mobile device, a portable computer, a desktop computer, and/or other like devices. Additionally or alternatively, user device 106 may include a device capable of receiving information from and/or communicating information to other user devices (e.g., directly via wired or wireless communication connection, indirectly via communication network 108, and/or the like). In some non-limiting embodiments or aspects, user device 106 may be part of model management system 102 and/or part of the same system as model management system 102. For example, model management system 102, ML model management database 104, and user device 106 may all be (and/or be part of) a single system and/or a single computing device.

[0080] Communication network 108 may include one or more wired and/or wireless networks. For example, communication network 108 may include a cellular network (e.g., a long-term evolution (LTE) network, a third-generation (3G) network, a fourth-generation (4G) network, a fifth-generation (5G) network, a code division multiple access (CDMA) network, etc.), a public land mobile network (PLMN), a local area network (LAN), a wide area network (WAN), a metropolitan area network (MAN), a telephone network (e.g., the public switched telephone network (PSTN) and/or the like), a private network, an ad hoc network, an intranet, the Internet, a fiber optic-based network, a cloud computing network, and/or the like, and/or a combination of some or all of these or other types of networks.

[0081] The number and arrangement of systems and devices shown in FIG. 1 are provided as an example. There may be additional systems and/or devices, fewer systems and/or devices, different systems and/or devices, and/or differently arranged systems and/or devices than those shown in FIG. 1. Furthermore, two or more systems or devices shown in FIG. 1 may be implemented within a single system or device, or a single system or device shown in FIG. 1 may be implemented as multiple, distributed systems or devices. Additionally or alternatively, a set of systems (e.g., one or more systems) or a set of devices (e.g., one or more devices) of system 100 may

perform one or more functions described as being performed by another set of systems or another set of devices of system 100.

[0082] Referring now to FIG. 2, shown is a flow diagram for process 200 for operating a large scale graph transformer machine learning model network architecture, according to some non-limiting embodiments or aspects. In some non-limiting embodiments or aspects, one or more of the steps of process 200 may be performed (e.g., completely, partially, etc.) by model management system 102 (e.g., one or more devices of model management system 102). In some non-limiting embodiments or aspects, one or more of the steps of process 200 may be performed (e.g., completely, partially, etc.) by another device or a group of devices separate from or including model management system 102 (e.g., one or more devices of model management system 102), ML model management database 104, and/or user device 106. The steps shown in FIG. 2 are for example purposes only. It will be appreciated that additional, fewer, different, and/or a different order of steps may be used in some non-limiting embodiments or aspects. In some non-limiting embodiments or aspects, a step may be automatically performed in response to performance and/or completion of a prior step.

[0083] As shown in FIG. 2, at step 202, process 200 includes receiving graph data associated with one or more graphs for a plurality of entities. For example, model management system 102 may receive the graph data associated with one or more graphs for the plurality of entities from ML model management database 104, user device 106, and/or another system or device. In some non-limiting embodiments or aspects, model management system 102 may receive a dataset (e.g., a training dataset) that includes graph data associated with a graph (e.g., a mathematical structure used to model pairwise relations between objects). In some non-limiting embodiments or aspects, the graph may include a plurality of nodes and a plurality of edges. In some non-limiting embodiments or aspects, the graph data may include node data associated with each node (e.g., vertex) and/or edge data associated with each edge (e.g., links) of the graph. In some non-limiting embodiments or aspects, each entity of the plurality of entities may be represented by a node of the graph.

[0084] In some non-limiting embodiments or aspects, the dataset may include graph data associated with a set of nodes (e.g., a set of at least 5, 10, 15, 30, 50, 100, 200, 300, etc., or more nodes). In some non-limiting embodiments or aspects, the

dataset may include graph data associated with a set of labeled nodes and/or a set of unlabeled nodes.

[0085] In some non-limiting embodiments or aspects, the graph data may include a plurality of node embeddings associated with a number of nodes in the graph and node data associated with each node of the graph. The node data may include data associated with parameters of each node in the graph. Additionally or alternatively, the node data may include entity data associated with a plurality of entities (e.g., a first entity, a second entity, etc.). The plurality of node embeddings may include a set of node embeddings that may be based on the entity data.

[0086] In some non-limiting embodiments or aspects, the graph data may be associated with a population of entities (e.g., users, accountholders, merchants, issuers, items provided by an entity, etc.) and includes a plurality of data instances associated with a plurality of features. In some non-limiting embodiments or aspects, the plurality of data instances (e.g., represented as edge data associated with each edge of the graph) of the graph data may represent a plurality of interactions (e.g., transactions, such as electronic payment transactions) conducted by the population. In some examples, the graph data may include a large amount of data instances, such as 100 data instances, 500 data instances, 1,000 data instances, 5,000 data instances, 10,000 data instances, 25,000 data instances, 50,000 data instances, 100,000 data instances, 1,000,000 data instances, and/or the like.

[0087] In some non-limiting embodiments or aspects, each data instance may include transaction data associated with the transaction. In some non-limiting embodiments or aspects, the transaction data may include a plurality of transaction parameters associated with an electronic payment transaction. In some non-limiting embodiments or aspects, the plurality of features may represent the plurality of transaction parameters. In some non-limiting embodiments or aspects, the plurality of transaction parameters may include electronic wallet card data associated with an electronic card (e.g., an electronic credit card, an electronic debit card, an electronic loyalty card, and/or the like), decision data associated with a decision (e.g., a decision to approve or deny a transaction authorization request), authorization data associated with an authorization response (e.g., an approved spending limit, an approved transaction value, and/or the like), a PAN, an authorization code (e.g., a personal identification number (PIN), etc.), data associated with a transaction amount (e.g., an approved limit, a transaction value, etc.), data associated with a transaction date and

time, data associated with a conversion rate of a currency, data associated with a merchant type (e.g., a merchant category code that indicates a type of goods, such as grocery, fuel, and/or the like), data associated with an acquiring institution country, data associated with an identifier of a country associated with the PAN, data associated with a response code, data associated with a merchant identifier (e.g., a merchant name, a merchant location, and/or the like), data associated with a type of currency corresponding to funds stored in association with the PAN, and/or the like.

[0088] In some non-limiting embodiments or aspects, model management system 102 may receive a dataset that includes graph data associated with a first graph for a first entity and graph data associated with a second graph for a second entity. The first graph may include a first plurality of nodes and edges and/or the second graph may include a second plurality of nodes and edges. In some non-limiting embodiments or aspects, the first entity may be represented by a first node (e.g., a first target node) of the first graph and/or the second entity may be represented by a second node (e.g., a second target node) of the second graph. In some non-limiting embodiments or aspects, the first graph may be the same as or similar to the second graph.

[0089] In some non-limiting embodiments or aspects, model management system 102 may generate the graph data associated with a graph. For example, model management system 102 may perform a first search (e.g., a first breadth first search) of the first graph for the first entity to obtain the graph data associated with the first graph and/or perform a second search (e.g., a second breadth first search) of the second graph for the second entity to obtain the graph data associated with the second graph.

[0090] As shown in FIG. 2, at step 204, process 200 includes aggregating information from adjacent nodes and associated edges into node representations. For example, model management system 102 may aggregate information (e.g., neighborhood information of a node, such as edge data associated with edges of the graph and/or node data associated with nodes of the graph) from adjacent nodes and associated edges of one or more graphs into node representations. In some non-limiting embodiments or aspects, a node representation may include a vector representation of the information. In some non-limiting embodiments or aspects, model management system 102 may aggregate information from adjacent nodes and associated edges of a first graph for a first entity into a first plurality of node

representations and/or aggregate information from adjacent nodes and associated edges of a second graph for a second entity into a second plurality of node representations.

[0091] In some non-limiting embodiments or aspects, model management system 102 may aggregate k-hop neighborhood information of a target node from associated nodes and associated edges of one or more graphs into a node representation of the target node. For example, model management system 102 may aggregate neighborhood information of a target node from associated nodes and associated edges that are within k-hops of the target node into a node representation for the target node. In some non-limiting embodiments or aspects, model management system 102 may determine neighborhood information of a target node from associated nodes and/or associated edges based on a number of hops between the target node and an associated node and/or an associated edge. In some non-limiting embodiments or aspects, model management system 102 may transform the neighborhood information for each hop (e.g., 1 hop, 2 hops, 3 hops, etc.) into a node representation for the target node.

[0092] In some non-limiting embodiments or aspects, model management system 102 may generate one or more GNNs based on the graph data associated with one or more graphs for a plurality of entities. For example, model management system 102 may generate a first GNN based on graph data associated with the first graph for the first entity and generate a second GNN based on graph data associated with a second graph for a second entity. In some non-limiting embodiments or aspects, a GNN (e.g., the first GNN or the second GNN) may include a plurality of positional embeddings that represent position data for the position of each node (e.g., the position of each node with regard to a number of hops (e.g., based on edges) between each node) of the plurality of nodes in a graph (e.g., a graph upon which the GNN is based). Additionally or alternatively, a GNN may include a plurality of node embeddings that represent node data for each node of the plurality of nodes of the graph.

[0093] In some non-limiting embodiments or aspects, when aggregating information from adjacent nodes and associated edges of a graph into a plurality of node representations, model management system 102 may combine embeddings (e.g., positional embeddings or node embeddings) of a GNN to provide the plurality of node representations. For example, when aggregating the information from adjacent nodes and associated edges of the first graph into the first plurality of node

representations, model management system 102 may concatenate a first node embedding of the first GNN and a first positional embedding of the first GNN to provide the first plurality of node representations. In some non-limiting embodiments or aspects, when aggregating the information from adjacent nodes and associated edges of the second graph into the second plurality of node representations, model management system 102 may concatenate a second node embedding of the second graph neural network and a second positional embedding of the second graph neural network to provide the second plurality of node representations.

[0094] As shown in FIG. 2, at step 206, process 200 includes generating sequence data associated with one or more sequences. For example, model management system 102 may generate sequence data associated with one or more sequences based on a plurality of node representations. In some non-limiting embodiments or aspects, model management system 102 may generate the sequence data based on all nodes of a graph. For example, model management system 102 may generate first sequence data based on all nodes of a first graph associated with a first entity and second sequence data based on all nodes of a second graph associated with a second entity. In some non-limiting embodiments or aspects, the sequence data associated with a sequence may include a plurality of tokens (e.g., a plurality of token vectors). For example, the sequence data associated with a sequence may include a plurality of tokens based on a target node (e.g., a target node associated with a first entity, a target node associated with a second entity). In some non-limiting embodiments or aspects, the sequence data associated with a sequence may include a plurality of tokens based on neighborhood information of a target node. For example, the sequence data associated with a sequence may include a plurality of tokens that are in order based on k-hop neighborhood information of a target node. In some non-limiting embodiments or aspects, model management system 102 may generate a sequence for each node of a graph, which incorporates tokens from different hops to preserve the neighborhood information.

[0095] In some non-limiting embodiments or aspects, model management system 102 may generate first sequence data associated with a first sequence of the first plurality of node representations and/or generate second sequence data associated with a second sequence of the second plurality of node representations.

[0096] As shown in FIG. 2, at step 208, process 200 includes determining a prediction of a relationship between entities based on the sequence data. For

example, model management system 102 may determine the prediction of a relationship between entities based on the sequence data.

[0097] In some non-limiting embodiments or aspects, model management system 102 may provide sequence data associated with one or more sequences and/or as an input to a transformer machine learning model (e.g., a transformer block). In some non-limiting embodiments or aspects, the transformer machine learning model comprises a plurality of transformer layers. Each transformer layer may include a multi-head self-attention (MSA) network or single-head self-attention (SSA) network and a position-wise feed-forward network (FFN). In some non-limiting embodiments or aspects, the transformer machine learning model may treat each node as a sequence of tokens, which allows for training the transformer machine learning model in a mini-batch manner, and this allows for the transformer machine learning model to handle graphs with large amounts of data using less computational resources than a comparable machine learning model that does not follow treatment of each node as a sequence of tokens.

[0098] In some non-limiting embodiments or aspects, model management system 102 may provide the first sequence data associated with the first sequence and/or the second sequence data associated with the second sequence as an input to a transformer machine learning model (e.g., a block of a transformer machine learning model) and determine a prediction of a relationship between the first entity and the second entity based on an output of the transformer machine learning model. In some non-limiting embodiments or aspects, the output of the transformer machine learning model may include a final embedding. For example, the output of the transformer machine learning model may include a first final embedding associated with a first entity and a second final embedding associated with a second entity.

[0099] In some non-limiting embodiments or aspects, model management system 102 may generate a score (e.g., a preference score, a prediction score, etc.) based on a final embedding provided by the transformer machine learning model. For example, model management system 102 may generate the score based on a first final embedding associated with a first entity and a second final embedding associated with a second entity.

[0100] In some non-limiting embodiments or aspects, model management system 102 may calculate an inner product of a first final embedding associated with the first entity and a second final embedding associated with the second entity to provide the

score. In some non-limiting embodiments or aspects, model management system 102 may determine a prediction of a relationship between entities (e.g., between the first entity and the second entity) based on the score.

[0101] In some non-limiting embodiments or aspects, model management system 102 may calculate an inner product of a first final embedding associated with the first entity and a second final embedding associated with the second entity and, when determining the prediction of a relationship between the first entity and the second entity, model management system 102 may determine the prediction of a relationship between the first entity and the second entity based on the inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity. In some non-limiting embodiments or aspects, model management system 102 may generate a recommendation for the first entity and/or the second entity based on the prediction of a relationship between the first entity and the second entity. In some non-limiting embodiments or aspects, the first entity may include a consumer and the second entity may include an item (e.g., a good or service provided by a merchant, an account provided by an issuer, a good or service associated with an account provided by an issuer, etc.). In such an example, model management system 102 may generate a recommendation for the consumer that is associated with the item based on the prediction of a relationship between the consumer and the item.

[0102] In some non-limiting embodiments or aspects, model management system 102 may perform an action, such as a fraud prevention procedure, a transaction authorization procedure, and/or a recommendation procedure based on the prediction of a relationship between entities. For example, model management system 102 may perform the action based on determining to perform the action. In some non-limiting embodiments or aspects, model management system 102 may perform a fraud prevention procedure associated with protection of an account of a user (e.g., a first entity, such as a user associated with user device 106) based on an output of the transformer machine learning model and/or the prediction of a relationship between entities (e.g., a prediction of a relationship between a first entity and a second entity). For example, if the output of the transformer machine learning model and/or the prediction of a relationship between entities indicates that the fraud prevention procedure is necessary, model management system 102 may perform the fraud prevention procedure associated with protection of the account of the user. In such an

example, if the output of the transformer machine learning model and/or the prediction of a relationship between entities indicates that the fraud prevention procedure is not necessary, model management system 102 may forego performing the fraud prevention procedure associated with protection of the account of the user.

[0103] Referring now to FIGS. 3A-3E, shown are schematic diagrams of implementation 300 of a process (e.g., process 200) for operating a large scale graph transformer machine learning model network architecture. In some non-limiting embodiments or aspects, one or more of the steps of the process may be performed (e.g., completely, partially, etc.) by model management system 102 (e.g., one or more devices of model management system 102). In some non-limiting embodiments or aspects, one or more of the steps of the process may be performed (e.g., completely, partially, etc.) by another device or a group of devices separate from or including model management system 102 (e.g., one or more devices of model management system 102), ML model management database 104, and/or user device 106.

[0104] As shown by reference number 305 in FIG. 3A, model management system 102 may receive a first graph for a first entity and a second graph for a second entity from ML model management database 104. In some non-limiting embodiments or aspects, the first graph data may represent a set of first entities as a set of users, $\mathcal{U} = \{u\}$, and the second graph data may represent a set of second entities as a set of items, $I = \{i\}$, where I_u^+ represents items that a user, u, has interacted with before, while $I_u^- = I - I_u^+$ represents items that have not been observed by u. The user-item interactions may be represented as a bipartite graph, $\mathcal{G}$, and a user-item rating matrix $\mathbf{R} \in \mathbb{R}^{|\mathcal{U}| x |I|}$ where $|\mathcal{U}|$ and $|I|$ denote the number of users and items, respectively. Each entry is equal to 1, if user u has interacted with item i, and is equal to 0 otherwise. The corresponding adjacency matrix $\mathbf{A}$ for the bipartite graph can be obtained as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{bmatrix}$$

[0105]

[0106] As further shown by reference number 310 in FIG. 3A, model management system 102 may generate first graph data associated with the first graph for the first entity and second graph data associated with the second graph for the second entity. For example, model management system 102 may perform a first search (e.g., a first

breadth first search) of the first graph for the first entity to obtain the graph data associated with the first graph and/or perform a second search (e.g., a second breadth first search) of the second graph for the second entity to obtain the graph data associated with the second graph.

[0107] As further shown by reference number 315 in FIG. 3B, model management system 102 may aggregate information of the first graph into first node representations and aggregate information of the second graph into second node representations. In some non-limiting embodiments or aspects, a node representation may include a vector representation of the information. In some non-limiting embodiments or aspects, model management system 102 may aggregate information from adjacent nodes and associated edges of a first graph for a first entity into a first plurality of node representations and/or aggregate information from adjacent nodes and associated edges of a second graph for a second entity into a second plurality of node representations.

[0108] In some non-limiting embodiments or aspects, model management system 102 may aggregate k-hop neighborhood information of a target node from associated nodes and associated edges of one or more graphs into a node representation of the target node. For example, model management system 102 may aggregate neighborhood information of a target node from associated nodes and associated edges that are within k-hops of the target node into a node representation for the target node. In some non-limiting embodiments or aspects, model management system 102 may determine neighborhood information of a target node from associated nodes and/or associated edges based on a number of hops between the target node and an associated node and/or an associated edge. In some non-limiting embodiments or aspects, model management system 102 may transform the neighborhood information for each hop (e.g., 1 hop, 2 hops, 3 hops, etc.) into a node representation for the target node.

[0109] In some non-limiting embodiments or aspects, model management system 102 may obtain a positional embedding and a node embedding for each of the first and second graphs based on the first graph data and the second graph data, respectively. In some non-limiting embodiments or aspects, model management system 102 may obtain a first node embedding associated with the first entity for the first graph and a second node embedding associated with the second entity for the

second graph based on a lookup process (e.g., a lookup process with an embedding lookup table):

[0110] $$\mathbf{E}_{id} = \text{lookup}(ID)$$

[0111] Additionally or alternatively, model management system 102 may obtain a first positional embedding associated with the first entity for the first graph and a second positional embedding associated with the second entity for the second graph based on structural data (e.g., positional data associated with a position of a node) associated with the first graph data and the second graph data, respectively.

[0112] In some non-limiting embodiments or aspects, model management system 102 may concatenate a node embedding and a positional embedding to provide the plurality of node representations. For example, model management system 102 may concatenate a first node embedding and a first positional embedding to provide a first plurality of node representations for a first entity and concatenate a second node embedding and a second positional embedding to provide a second plurality of node representations for a second entity.

[0113] In some non-limiting embodiments or aspects, signature vectors of a Laplacian matrix of the first graph and the second graph may be used to determine the structural data of nodes of the first graph and the second graph:

[0114] $$\mathbf{E}_{\text{user}} = \mathbf{E}_{\text{user}} \| \mathbf{U}, \mathbf{E}_{\text{item}} = \mathbf{E}_{\text{item}} \| \mathbf{V}, \quad \text{, where, } \mathbf{U}, \mathbf{\Sigma}, \mathbf{V} = SVD(\mathbf{R})$$

[0115] where || indicates the concatenation operator, *SVD* is the singular value decomposition of user-item rating matrix **R**, and U, ∑, and V are resultant vectors of the *SVD*.

[0116] For a node, v, the k-hop neighborhood information may be represented as:

[0117] $$\mathcal{N}^k(v) = \{u \in V | d(v,u) \leq k\}$$

[0118] Where $d(v,u)$ represents the shortest distance between node v and node u. $\mathcal{N}^0(v) = \{v\}$ is defined as the 0-hop neighborhood and is the node v itself. The k-hop neighborhood $\mathcal{N}^k(v)$ may be transformed into a node embedding, $\mathbf{e}_v^k$, (e.g., an embedding that represents the nodes within a predetermined number of hops of a target node, a neighborhood embedding, etc.) with an aggregation operator, ϕ.

[0119] With this, the node representation (e.g., the k-hop representation) of a node v can be expressed as:

[0120] $$\mathbf{e}_v^k = \phi(\mathcal{N}^k(v))$$

[0121] As further shown by reference number 320 in FIG. 3C, model management system 102 may generate first sequence data associated with a first sequence and second sequence data associated with a second sequence. In some non-limiting embodiments or aspects, model management system 102 may generate the sequence data based on all nodes of a graph. For example, model management system 102 may generate first sequence data based on all nodes of a first graph associated with a first entity and second sequence data based on all nodes of a second graph associated with a second entity. In some non-limiting embodiments or aspects, the sequence data associated with a sequence may include a plurality of tokens (e.g., a plurality of token vectors). For example, the sequence data associated with a sequence may include a plurality of tokens based on a target node (e.g., a target node associated with a first entity, a target node associated with a second entity). In some non-limiting embodiments or aspects, the sequence data associated with a sequence may include a plurality of tokens based on neighborhood information of a target node. For example, the sequence data associated with a sequence may include a plurality of tokens that are in order based on k-hop neighborhood information of a target node. In some non-limiting embodiments or aspects, model management system 102 may generate a sequence for each node of a graph, which incorporates tokens from different hops to preserve the neighborhood information.

[0122] In some non-limiting embodiments or aspects, model management system 102 may generate first sequence data associated with a first sequence of the first plurality of node representations and/or generate second sequence data associated with a second sequence of the second plurality of node representations.

[0123] In some non-limiting embodiments or aspects, model management system 102 may calculate the neighborhood embeddings for k-hops of a node and further construct a sequence, S_v, to represent the neighborhood information of node v as:

[0124] $$S_v = (\mathbf{e}_v^0, \mathbf{e}_v^1, ..., \mathbf{e}_v^K)$$

[0125] where K is a hyperparameter. $\mathbf{e}_v^k$ may represent a d-dimensional vector, and the sequences of all nodes in a graph may be used by model management system 102 to construct a tensor:

[0126] $$\mathbf{E} \in \mathbb{R}^{n \times (K+1) \times d}$$

[0127] In some non-limiting embodiments or aspects, $\mathbf{E}$ may be decomposed to a sequence (e.g., a first sequence associated with the first entity or a second sequence associated with the second entity) S:

[0128] $$S = (\mathbf{E}_0, \mathbf{E}_1, \cdots, \mathbf{E}_K)$$

[0129] where $\mathbf{E}_k \in \mathbb{R}^{n \times d}$ may refer to the k-hop neighborhood matrix and $\mathbf{E}_0$ may refer to a first node embedding associated with the first entity for the first graph and/or a second node embedding associated with the second entity for the second graph that was obtained based on a lookup process as described above.

[0130] In some non-limiting embodiments or aspects, model management system 102 may obtain a sequence (e.g., a first sequence associated with the first entity or a second sequence associated with the second entity) of k-hop neighborhood matrices by applying a propagation process to the first graph and/or the second graph. In some non-limiting embodiments or aspects, a k-hop neighborhood matrix may be described as:

[0131] $$\mathbf{E}_k = \hat{\mathbf{A}}^k \mathbf{E}$$

[0132] As further shown by reference number 325 in FIG. 3D, model management system 102 may generate a first final embedding and a second final embedding. In some non-limiting embodiments or aspects, model management system 102 may provide sequence data associated with one or more sequences and/or as an input to a transformer machine learning model (e.g., a transformer block). In some non-limiting embodiments or aspects, the transformer machine learning model comprises a plurality of transformer layers. Each transformer layer may include a MSA network or SSA network and an FFN.

[0133] In some non-limiting embodiments or aspects, model management system 102 may provide the first sequence data associated with the first sequence and/or the second sequence data associated with the second sequence as an input to the transformer block. In some non-limiting embodiments or aspects, the output of the transformer machine learning model may include a final embedding. For example, the output of the transformer machine learning model may include the first final embedding associated with the first entity and the second final embedding associated with the second entity. In some non-limiting embodiments or aspects, model management system 102 may determine a prediction of a relationship between the first entity and the second entity based on an output of the transformer machine learning model.

[0134] In some non-limiting embodiments or aspects, model management system 102 may provide an input (e.g., an input that includes first sequence data for a first sequence and second sequence data for a second sequence) of $\mathbf{E}_k \in \mathbb{R}^{n \times d}$ to a self-attention network of the transformer block, where n is a number of tokens and d is a hidden dimension. In some non-limiting embodiments or aspects, the transformer block may include a transformer encoder machine learning model. In some non-limiting embodiments or aspects, the transformer block may include a plurality of transformer layers. Each transformer layer may include a MSA network or SSA network and an FFN.

[0135] In some non-limiting embodiments or aspects, the self-attention network may process the information by projecting $\mathbf{E}_k$ into three subspaces, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$:

[0136] $$\mathbf{Q} = \mathbf{E}_k \mathbf{W}^Q,\ \mathbf{K} = \mathbf{E}_k \mathbf{W}^K,\ \mathbf{V} = \mathbf{E}_k \mathbf{W}^V$$

[0137] where $\mathbf{W}^Q \in \mathbb{R}^{d \times d_K}, \mathbf{W}^K \in \mathbb{R}^{d \times d_K}$ and $\mathbf{W}^V \in \mathbb{R}^{d \times d_V}$ are the projection matrices. An output of the self-attention network may be calculated as:

[0138] $$\mathbf{E}' = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_K}}\right)\mathbf{V}$$

[0139] The attention matrix, $\mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_K}}\right)$, may capture the pair-wise similarity of input tokens in a sequence (e.g., a first sequence associated with the first entity or a second sequence associated with the second entity). For example, the attention matrix may be used to calculate the dot product between each token pair after a projection procedure with regard to the transformer block. In some non-limiting embodiments or aspects, the softmax operation may be applied row-wise to the attention matrix.

[0140] In some non-limiting embodiments or aspects, an FFN of a transformer layer may include two linear layers with a Gaussian Error Linear Unit (GELU) non-linearity, and a LayerNorm (LN) is applied before each of the MSA and FFN:

[0141] $$\mathbf{E}_v^{(\ell)} = \mathrm{MSA}\left(\mathrm{LN}\left(\mathbf{E}_v^{(\ell-1)}\right)\right) + \mathbf{E}_v^{(\ell-1)},$$
$$\mathbf{E}_v^{(\ell)} = \mathrm{FFN}\left(\mathrm{LN}\left(\mathbf{E}_v^{(\ell)}\right)\right) + \mathbf{E}_v^{(\ell)},$$

[0142] where $\ell = 1, \ldots, L$ implies the ℓ-th layer of the transformer block. In some non-limiting embodiments or aspects, model management system 102 may apply a readout function to an output of the transformer block. Through a plurality of

transformer layers, a corresponding output, $\mathbf{Z}_v^{(\ell)}$, includes embeddings for all neighborhoods of node v. In some non-limiting embodiments or aspects, model management system 102 may apply the readout function to $\mathbf{Z}_v^{(\ell)}$ to aggregate information of different neighborhoods into the final embedding for each of the first entity and the second entity. In some non-limiting embodiments or aspects, the readout function may include a summation function and/or a mean function.

[0143] In some non-limiting embodiments or aspects, by propagating the L layer, model management system 102 may obtain $L+1$ embeddings to represent a first entity (e.g., a user) as:

[0144] $(\mathbf{e}_u^{(0)}, \ldots, \mathbf{e}_u^{(L)})$

[0145] and $L+1$ embeddings to represent a second entity (e.g., an item) as:

[0146] $(\mathbf{e}_i^{(0)}, \ldots, \mathbf{e}_i^{(L)})$

[0147] In some non-limiting embodiments or aspects, an aggregation function may be used to obtain the final embeddings for the first entity and the second entity:

[0148] $\mathbf{e}_u^* = \mathrm{AGG}(\mathbf{e}_u^{(0)}, \ldots, \mathbf{e}_u^{(L)}), \quad \mathbf{e}_i^* = \mathrm{AGG}(\mathbf{e}_i^{(0)}, \ldots, \mathbf{e}_i^{(L)})$

[0149] In some non-limiting embodiments or aspects, the aggregation function may include a weighted sum aggregation. In some non-limiting embodiments or aspects, model management system 102 may use an inner product to predict a score (e.g., a preference score), $\hat{y}_{ui}$ between the first entity and the second entity:

[0150] $\hat{y}_{ui} = \mathbf{e}_u^{*T} \mathbf{e}_i^*$

[0151] In some non-limiting embodiments or aspects, model management system 102 may determine a prediction of a relationship between the first entity and the second entity based on the score. For example, if the score satisfies a threshold value, then model management system 102 may determine that there is a prediction of a relationship between the first entity and the second entity. In such an example, if the score does not satisfy a threshold value, then model management system 102 may determine that there is not a prediction of a relationship between the first entity and the second entity. In some non-limiting embodiments or aspects, model management system 102 may use a loss function, such as a Bayesian Personalized Ranking (BPR) loss function, to optimize the model parameters of the transformer block, which is used to minimize the following:

[0152] $$\mathcal{L}_{bpr}(\Theta) = \sum_{(u,i,j)\in\mathcal{O}} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \alpha\|\Theta\|_2^2$$

[0153] where:

[0154] $$\mathcal{O} = \{(u,i,j) \mid u \in \mathcal{U} \wedge i \in I_u^+ \wedge j \in I_u^-\}$$

[0155] denotes the pairwise training data, $\sigma(\cdot)$ is the sigmoid function, Θ denotes the model parameters, and α controls the L_2 norm to prevent over-fitting.

[0156] As further shown by reference number 330 in FIG. 3E, model management system 102 may generate a recommendation for the first entity. For example, model management system 102 may generate the recommendation for the first entity based on the first final embedding and the second final embedding. In some non-limiting embodiments or aspects, model management system 102 may generate the recommendation for the first entity based on a prediction of a relationship between the first entity and the second entity. In some non-limiting embodiments or aspects, model management system 102 may calculate an inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity and, when determining the prediction of a relationship between the first entity and the second entity, model management system 102 may determine the prediction of a relationship between the first entity and the second entity based on the inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity.

[0157] As further shown by reference number 335 in FIG. 3E, model management system 102 may transmit the recommendation. For example, model management system 102 may transmit the recommendation to user device 106 e.g., a user, as the first entity, associated with user device 106 based on generating the recommendation for the first entity.

[0158] Referring now to FIG. 4, shown is a diagram of a non-limiting embodiment or aspect of exemplary environment 400 in which methods, systems, and/or products, as described herein, may be implemented. As shown in FIG. 4, environment 400 may include transaction service provider system 402, issuer system 404, customer device 406, merchant system 408, acquirer system 410, and communication network 412. In some non-limiting embodiments or aspects, each of model management system 102, ML model management database 104, and/or user device 106 of FIG. 1 may be implemented by (e.g., part of) transaction service provider system 402. In some non-

limiting embodiments or aspects, at least one of model management system 102, ML model management database 104, and/or user device 106 of FIG. 1 may be implemented by (e.g., part of) another system, another device, another group of systems, or another group of devices, separate from or including transaction service provider system 402, such as issuer system 404, customer device 406, merchant system 408, acquirer system 410, and/or the like.

[0159] Transaction service provider system 402 may include one or more devices capable of receiving information from and/or communicating information to issuer system 404, customer device 406, merchant system 408, and/or acquirer system 410 via communication network 412. For example, transaction service provider system 402 may include a computing device, such as a server (e.g., a transaction processing server), a group of servers, and/or other like devices. In some non-limiting embodiments or aspects, transaction service provider system 402 may be associated with a transaction service provider, as described herein. In some non-limiting embodiments or aspects, transaction service provider system 402 may be in communication with a data storage device, which may be local or remote to transaction service provider system 402. In some non-limiting embodiments or aspects, transaction service provider system 402 may be capable of receiving information from, storing information in, communicating information to, or searching information stored in the data storage device.

[0160] Issuer system 404 may include one or more devices capable of receiving information and/or communicating information to transaction service provider system 402, customer device 406, merchant system 408, and/or acquirer system 410 via communication network 412. For example, issuer system 404 may include a computing device, such as a server, a group of servers, and/or other like devices. In some non-limiting embodiments or aspects, issuer system 404 may be associated with an issuer institution, as described herein. For example, issuer system 404 may be associated with an issuer institution that issued a credit account, debit account, credit card, debit card, and/or the like to a user associated with customer device 406.

[0161] Customer device 406 may include one or more devices capable of receiving information from and/or communicating information to transaction service provider system 402, issuer system 404, merchant system 408, and/or acquirer system 410 via communication network 412. Additionally or alternatively, each customer device 406 may include a device capable of receiving information from and/or communicating

information to other customer devices 406 via communication network 412, another network (e.g., an ad hoc network, a local network, a private network, a virtual private network, and/or the like), and/or any other suitable communication technique. For example, customer device 406 may include a client device and/or the like. In some non-limiting embodiments or aspects, customer device 406 may or may not be capable of receiving information (e.g., from merchant system 408 or from another customer device 406) via a short-range wireless communication connection (e.g., an NFC communication connection, an RFID communication connection, a Bluetooth® communication connection, a Zigbee® communication connection, and/or the like), and/or communicating information (e.g., to merchant system 408) via a short-range wireless communication connection.

[0162] Merchant system 408 may include one or more devices capable of receiving information from and/or communicating information to transaction service provider system 402, issuer system 404, customer device 406, and/or acquirer system 410 via communication network 412. Merchant system 408 may also include a device capable of receiving information from customer device 406 via communication network 412, a communication connection (e.g., an NFC communication connection, an RFID communication connection, a Bluetooth® communication connection, a Zigbee® communication connection, and/or the like) with customer device 406, and/or the like, and/or communicating information to customer device 406 via communication network 412, the communication connection, and/or the like. In some non-limiting embodiments or aspects, merchant system 408 may include a computing device, such as a server, a group of servers, a client device, a group of client devices, and/or other like devices. In some non-limiting embodiments or aspects, merchant system 408 may be associated with a merchant, as described herein. In some non-limiting embodiments or aspects, merchant system 408 may include one or more client devices. For example, merchant system 408 may include a client device that allows a merchant to communicate information to transaction service provider system 402. In some non-limiting embodiments or aspects, merchant system 408 may include one or more devices, such as computers, computer systems, and/or peripheral devices capable of being used by a merchant to conduct a transaction with a user. For example, merchant system 408 may include a POS device and/or a POS system.

[0163] Acquirer system 410 may include one or more devices capable of receiving information from and/or communicating information to transaction service provider

system 402, issuer system 404, customer device 406, and/or merchant system 408 via communication network 412. For example, acquirer system 410 may include a computing device, a server, a group of servers, and/or the like. In some non-limiting embodiments or aspects, acquirer system 410 may be associated with an acquirer, as described herein.

[0164] Communication network 412 may include one or more wired and/or wireless networks. For example, communication network 412 may include a cellular network (e.g., a long-term evolution (LTE) network, a third generation (3G) network, a fourth generation (4G) network, a fifth generation (5G) network, a code division multiple access (CDMA) network, and/or the like), a public land mobile network (PLMN), a local area network (LAN), a wide area network (WAN), a metropolitan area network (MAN), a telephone network (e.g., the public switched telephone network (PSTN)), a private network (e.g., a private network associated with a transaction service provider), an ad hoc network, an intranet, the Internet, a fiber optic-based network, a cloud computing network, and/or the like, and/or a combination of these or other types of networks.

[0165] The number and arrangement of systems, devices, and/or networks shown in FIG. 4 are provided as an example. There may be additional systems, devices, and/or networks; fewer systems, devices, and/or networks; different systems, devices, and/or networks; and/or differently arranged systems, devices, and/or networks than those shown in FIG. 4. Furthermore, two or more systems or devices shown in FIG. 4 may be implemented within a single system or device, or a single system or device shown in FIG. 4 may be implemented as multiple, distributed systems or devices. Additionally or alternatively, a set of systems (e.g., one or more systems) or a set of devices (e.g., one or more devices) of environment 400 may perform one or more functions described as being performed by another set of systems or another set of devices of environment 400.

[0166] Referring now to FIG. 5, shown is a diagram of example components of device 500, according to non-limiting embodiments or aspects. Device 500 may correspond to at least one of model management system 102, ML model management database 104, and/or user device 106 in FIG. 1 and/or at least one of transaction service provider system 402, issuer system 404, customer device 406, merchant system 408, and/or acquirer system 410 in FIG. 4, as an example. In some non-limiting embodiments or aspects, such systems or devices in FIG. 1 or FIG. 4 may include at least one device 500 and/or at least one component of device 500. The number and

arrangement of components shown in FIG. 5 are provided as an example. In some non-limiting embodiments or aspects, device 500 may include additional components, fewer components, different components, or differently arranged components than those shown in FIG. 5. Additionally or alternatively, a set of components (e.g., one or more components) of device 500 may perform one or more functions described as being performed by another set of components of device 500.

[0167] As shown in FIG. 5, device 500 may include bus 502, processor 504, memory 506, storage component 508, input component 510, output component 512, and communication interface 514. Bus 502 may include a component that permits communication among the components of device 500. In some non-limiting embodiments or aspects, processor 504 may be implemented in hardware, firmware, or a combination of hardware and software. For example, processor 504 may include a processor (e.g., a central processing unit (CPU), a graphics processing unit (GPU), an accelerated processing unit (APU), etc.), a microprocessor, a digital signal processor (DSP), and/or any processing component (e.g., a field-programmable gate array (FPGA), an application-specific integrated circuit (ASIC), etc.) that can be programmed to perform a function. Memory 506 may include random access memory (RAM), read only memory (ROM), and/or another type of dynamic or static storage device (e.g., flash memory, magnetic memory, optical memory, etc.) that stores information and/or instructions for use by processor 504.

[0168] With continued reference to FIG. 5, storage component 508 may store information and/or software related to the operation and use of device 500. For example, storage component 508 may include a hard disk (e.g., a magnetic disk, an optical disk, a magneto-optic disk, a solid-state disk, etc.) and/or another type of computer-readable medium. Input component 510 may include a component that permits device 500 to receive information, such as via user input (e.g., a touch screen display, a keyboard, a keypad, a mouse, a button, a switch, a microphone, etc.). Additionally or alternatively, input component 510 may include a sensor for sensing information (e.g., a global positioning system (GPS) component, an accelerometer, a gyroscope, an actuator, etc.). Output component 512 may include a component that provides output information from device 500 (e.g., a display, a speaker, one or more light-emitting diodes (LEDs), etc.). Communication interface 514 may include a transceiver-like component (e.g., a transceiver, a separate receiver and transmitter, etc.) that enables device 500 to communicate with other devices, such as via a wired

connection, a wireless connection, or a combination of wired and wireless connections. Communication interface 514 may permit device 500 to receive information from another device and/or provide information to another device. For example, communication interface 514 may include an Ethernet interface, an optical interface, a coaxial interface, an infrared interface, a radio frequency (RF) interface, a universal serial bus (USB) interface, a Wi-Fi® interface, a cellular network interface, and/or the like.

[0169] Device 500 may perform one or more processes described herein. Device 500 may perform these processes based on processor 504 executing software instructions stored by a computer-readable medium, such as memory 506 and/or storage component 508. A computer-readable medium may include any non-transitory memory device. A memory device includes memory space located inside of a single physical storage device or memory space spread across multiple physical storage devices. Software instructions may be read into memory 506 and/or storage component 508 from another computer-readable medium or from another device via communication interface 514. When executed, software instructions stored in memory 506 and/or storage component 508 may cause processor 504 to perform one or more processes described herein. Additionally or alternatively, hardwired circuitry may be used in place of or in combination with software instructions to perform one or more processes described herein. Thus, embodiments described herein are not limited to any specific combination of hardware circuitry and software. The term “configured to,” as used herein, may refer to an arrangement of software, device(s), and/or hardware for performing and/or enabling one or more functions (e.g., actions, processes, steps of a process, and/or the like). For example, “a processor configured to” may refer to a processor that executes software instructions (e.g., program code) that cause the processor to perform one or more functions.

[0170] Although embodiments have been described in detail for the purpose of illustration, it is to be understood that such detail is solely for that purpose and that the disclosure is not limited to the disclosed embodiments or aspects, but, on the contrary, is intended to cover modifications and equivalent arrangements that are within the spirit and scope of the appended claims. For example, it is to be understood that the present disclosure contemplates that, to the extent possible, one or more features of any embodiment or aspect can be combined with one or more features of any other embodiment or aspect.

WHAT IS CLAIMED IS:

1. A computer-implemented method, comprising:

receiving, with at least one processor, graph data associated with a first graph for a first entity and graph data associated with a second graph for a second entity, wherein the first graph comprises a first plurality of nodes and edges, and wherein the second graph comprises a second plurality of nodes and edges;

aggregating, with at least one processor, information from adjacent nodes and associated edges of the first graph into a first plurality of node representations;

aggregating, with at least one processor, information from adjacent nodes and associated edges of the second graph into a second plurality of node representations;

generating, with at least one processor, first sequence data associated with a first sequence of the first plurality of node representations;

generating, with at least one processor, second sequence data associated with a second sequence of the second plurality of node representations;

providing, with at least one processor, the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs to a transformer machine learning model; and

determining, with at least one processor, a prediction of a relationship between the first entity and the second entity based on an output of the transformer machine learning model.

2. The computer-implemented method of claim 1, further comprising:

generating the output of the transformer machine learning model based on the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs, wherein the output comprises:

a first final embedding associated with the first entity, and

a second final embedding associated with the second entity.

3. The computer-implemented method of claim 2, further comprising:

calculating an inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity,

wherein determining the prediction of a relationship between the first entity and the second entity comprises:

determining the prediction of a relationship between the first entity and the second entity based on the inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity.

4. The computer-implemented method of claim 1, further comprising:

generating a first graph neural network based on the graph data associated with the first graph for the first entity; and

generating a second graph neural network based on the graph data associated with the second graph for the second entity.

5. The computer-implemented method of claim 4, wherein aggregating the information from adjacent nodes and associated edges of the first graph into the first plurality of node representations comprises:

concatenating a first node embedding of the first graph neural network and a first positional embedding of the first graph neural network to provide the first plurality of node representations, and

wherein aggregating the information from adjacent nodes and associated edges of the second graph into the second plurality of node representations comprises:

concatenating a second node embedding of the second graph neural network and a second positional embedding of the second graph neural network to provide the second plurality of node representations.

6. The computer-implemented method of claim 1, further comprising:

performing a first breadth first search of the first graph for the first entity to obtain the graph data associated with the first graph; and

performing a second breadth first search of the second graph for the second entity to obtain the graph data associated with the second graph.

7. The computer-implemented method of claim 1, further comprising:

generating a recommendation for the first entity based on the prediction of a relationship between the first entity and the second entity; and

transmitting the recommendation to a user device of the first entity.

8. A system, comprising:

at least one processor configured to:

receive graph data associated with a first graph for a first entity and graph data associated with a second graph for a second entity, wherein the first graph comprises a first plurality of nodes and edges, and wherein the second graph comprises a second plurality of nodes and edges;

aggregate information from adjacent nodes and associated edges of the first graph into a first plurality of node representations;

aggregate information from adjacent nodes and associated edges of the second graph into a second plurality of node representations;

generate first sequence data associated with a first sequence of the first plurality of node representations;

generate second sequence data associated with a second sequence of the second plurality of node representations;

provide the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs to a transformer machine learning model; and

determine a prediction of a relationship between the first entity and the second entity based on an output of the transformer machine learning model.

9. The system of claim 8, wherein the at least one processor is further configured to:

generate the output of the transformer machine learning model based on the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs, wherein the output comprises:

a first final embedding associated with the first entity, and

a second final embedding associated with the second entity.

10. The system of claim 9, wherein the at least one processor is further configured to:

calculate an inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity,

wherein, when determining the prediction of a relationship between the first entity and the second entity, the at least one processor is configured to:

determine the prediction of a relationship between the first entity and the second entity based on the inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity.

11. The system of claim 8, wherein the at least one processor is further configured to:

generate a first graph neural network based on the graph data associated with the first graph for the first entity; and

generate a second graph neural network based on the graph data associated with the second graph for the second entity.

12. The system of claim 11, wherein, when aggregating the information from adjacent nodes and associated edges of the first graph into the first plurality of node representations, the at least one processor is configured to:

concatenate a first node embedding of the first graph neural network and a first positional embedding of the first graph neural network to provide the first plurality of node representations, and

wherein, when aggregating the information from adjacent nodes and associated edges of the second graph into the second plurality of node representations, the at least one processor is configured to:

concatenate a second node embedding of the second graph neural network and a second positional embedding of the second graph neural network to provide the second plurality of node representations.

13. The system of claim 8, wherein the at least one processor is further configured to:

perform a first breadth first search of the first graph for the first entity to obtain the graph data associated with the first graph; and

perform a second breadth first search of the second graph for the second entity to obtain the graph data associated with the second graph.

14. The system of claim 8, wherein the at least one processor is further configured to:

generate a recommendation for the first entity based on the prediction of a relationship between the first entity and the second entity; and

transmit the recommendation to the first entity.

15. A computer program product comprising at least one non-transitory computer-readable medium including program instructions that, when executed by at least one processor, cause the at least one processor to:

receive graph data associated with a first graph for a first entity and graph data associated with a second graph for a second entity, wherein the first graph comprises a first plurality of nodes and edges, and wherein the second graph comprises a second plurality of nodes and edges;

aggregate information from adjacent nodes and associated edges of the first graph into a first plurality of node representations;

aggregate information from adjacent nodes and associated edges of the second graph into a second plurality of node representations;

generate first sequence data associated with a first sequence of the first plurality of node representations;

generate second sequence data associated with a second sequence of the second plurality of node representations;

provide the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs to a transformer machine learning model; and

determine a prediction of a relationship between the first entity and the second entity based on an output of the transformer machine learning model.

16. The computer program product of claim 15, wherein the program instructions further cause the at least one processor to:

generate the output of the transformer machine learning model based on the first sequence data associated with the first sequence and the second sequence data associated with the second sequence as inputs, wherein the output comprises:

a first final embedding associated with the first entity, and

a second final embedding associated with the second entity.

17. The computer program product of claim 16, wherein the program instructions further cause the at least one processor to:

calculate an inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity,

wherein the program instructions that cause the at least one processor to determine the prediction of a relationship between the first entity and the second entity, cause the at least one processor to:

determine the prediction of a relationship between the first entity and the second entity based on the inner product of the first final embedding associated with the first entity and the second final embedding associated with the second entity.

18. The computer program product of claim 15, wherein the program instructions further cause the at least one processor to:

generate a first graph neural network based on the graph data associated with the first graph for the first entity; and

generate a second graph neural network based on the graph data associated with the second graph for the second entity.

19. The computer program product of claim 18, wherein the program instructions that cause the at least one processor to aggregate the information from adjacent nodes and associated edges of the first graph into the first plurality of node representations, cause the at least one processor to:

concatenate a first node embedding of the first graph neural network and a first positional embedding of the first graph neural network to provide the first plurality of node representations, and

wherein the program instructions that cause the at least one processor to aggregate the information from adjacent nodes and associated edges of the second graph into the second plurality of node representations, cause the at least one processor to:

concatenate a second node embedding of the second graph neural network and a second positional embedding of the second graph neural network to provide the second plurality of node representations.

20. The computer program product of claim 15, wherein the program instructions further cause the at least one processor to:

perform a first breadth first search of the first graph for the first entity to obtain the graph data associated with the first graph; and

perform a second breadth first search of the second graph for the second entity to obtain the graph data associated with the second graph.

21. The computer program product of claim 15, wherein the program instructions further cause the at least one processor to:

generate a recommendation for the first entity based on the prediction of a relationship between the first entity and the second entity; and

transmit the recommendation to the first entity.

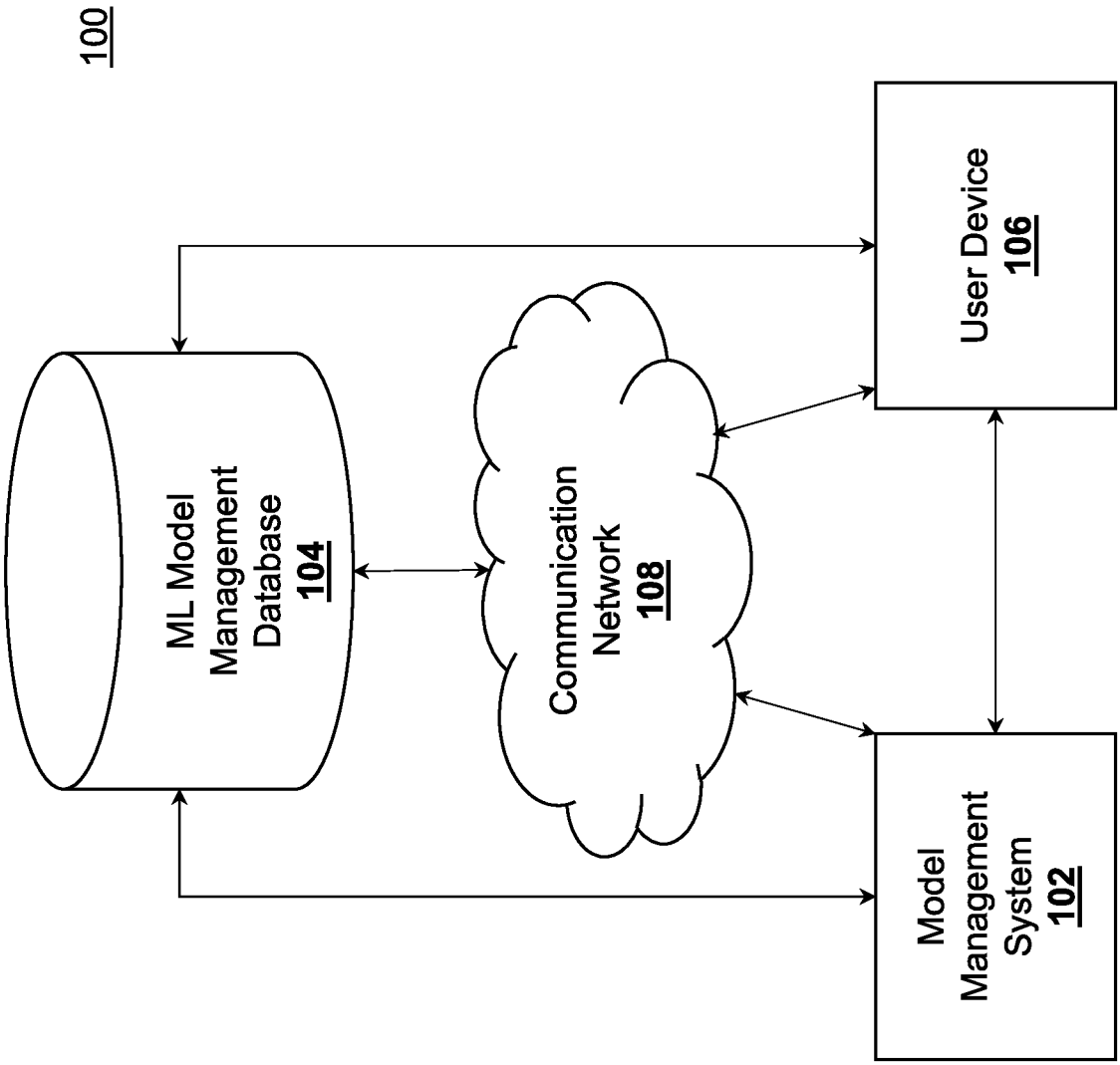
100
ML Model Management Database 104
Communication Network 108
User Device 106
Model Management System 102

FIG. 1

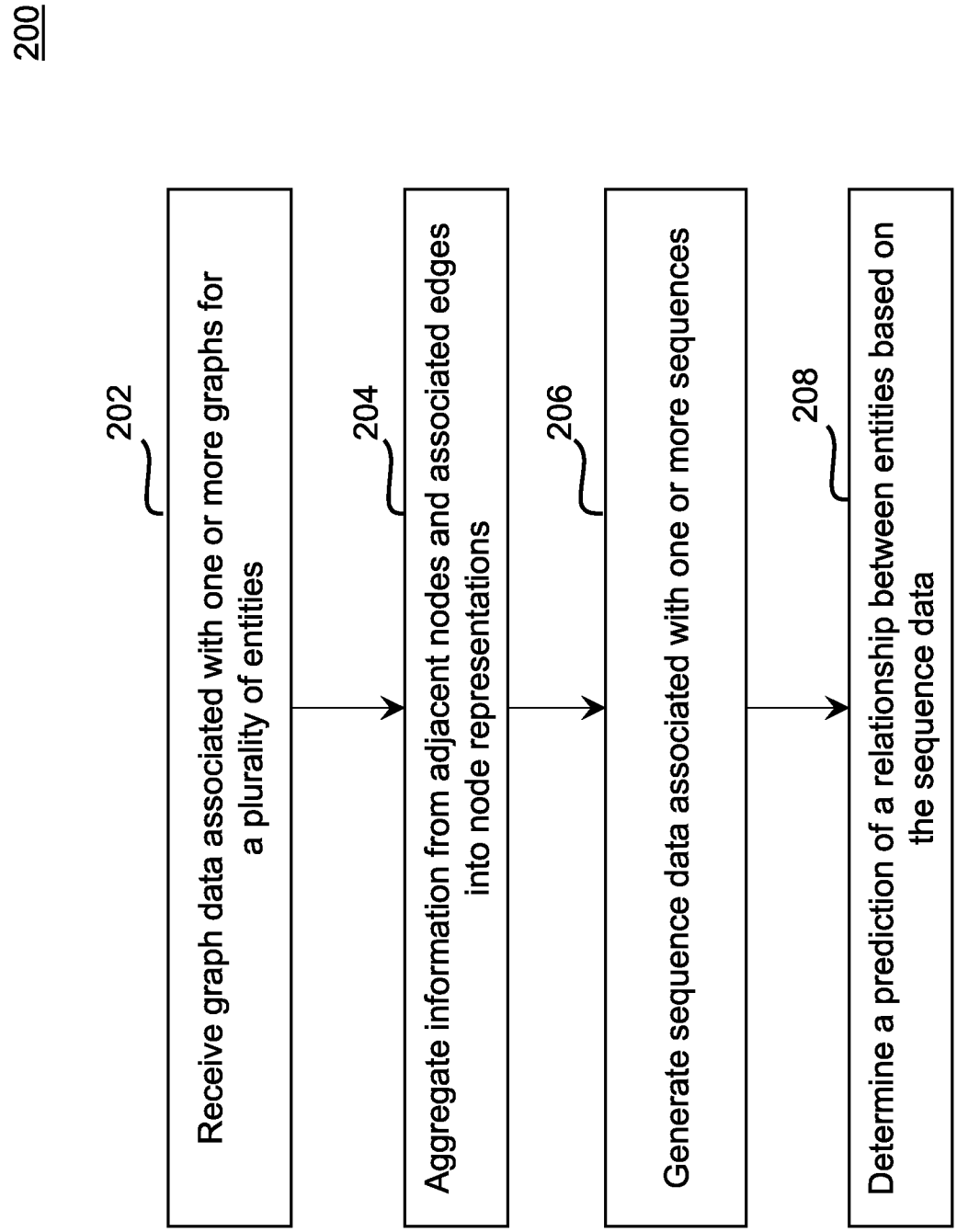
200
202
Receive graph data associated with one or more graphs for a plurality of entities
204
Aggregate information from adjacent nodes and associated edges into node representations
206
Generate sequence data associated with one or more sequences
208
Determine a prediction of a relationship between entities based on the sequence data

FIG. 2

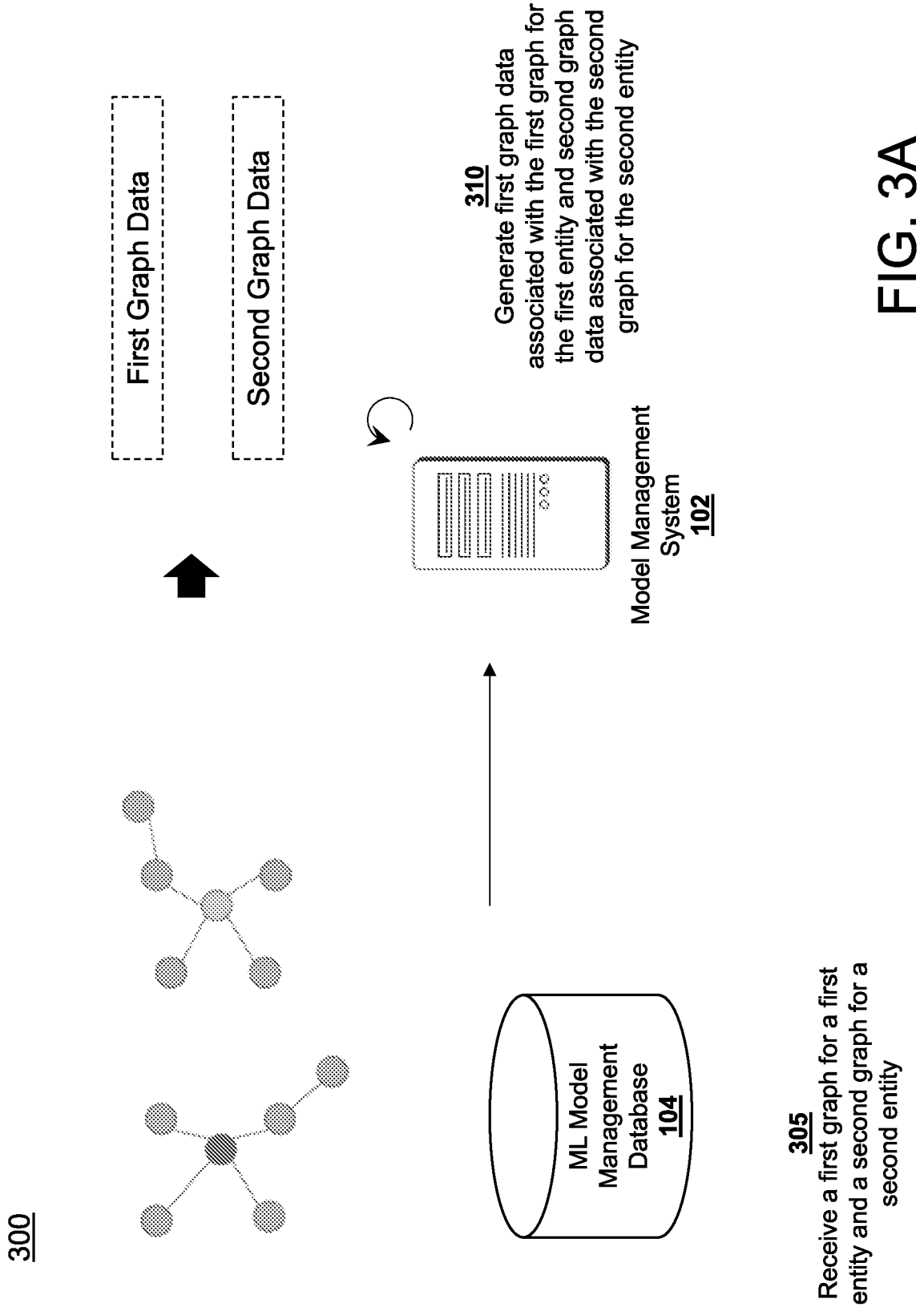
300
305
Receive a first graph for a first entity and a second graph for a second entity
ML Model Management Database
104
310
Generate first graph data associated with the first graph for the first entity and second graph data associated with the second graph for the second entity
Model Management System
102
First Graph Data
Second Graph Data

FIG. 3A

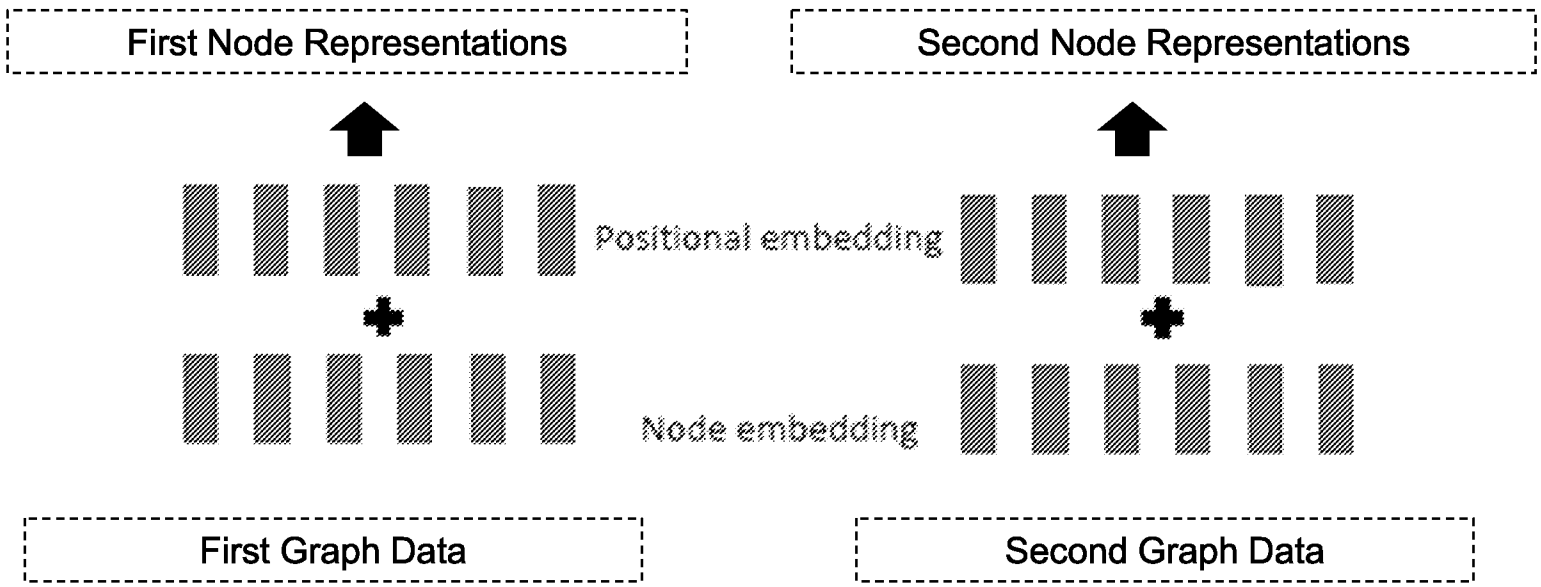
300
First Node Representations
Second Node Representations
Positional embedding
Node embedding
First Graph Data
Second Graph Data
Model Management System
102
315
Aggregate information of the first graph into first node representations and aggregate information of the second graph into second node representations

FIG. 3B

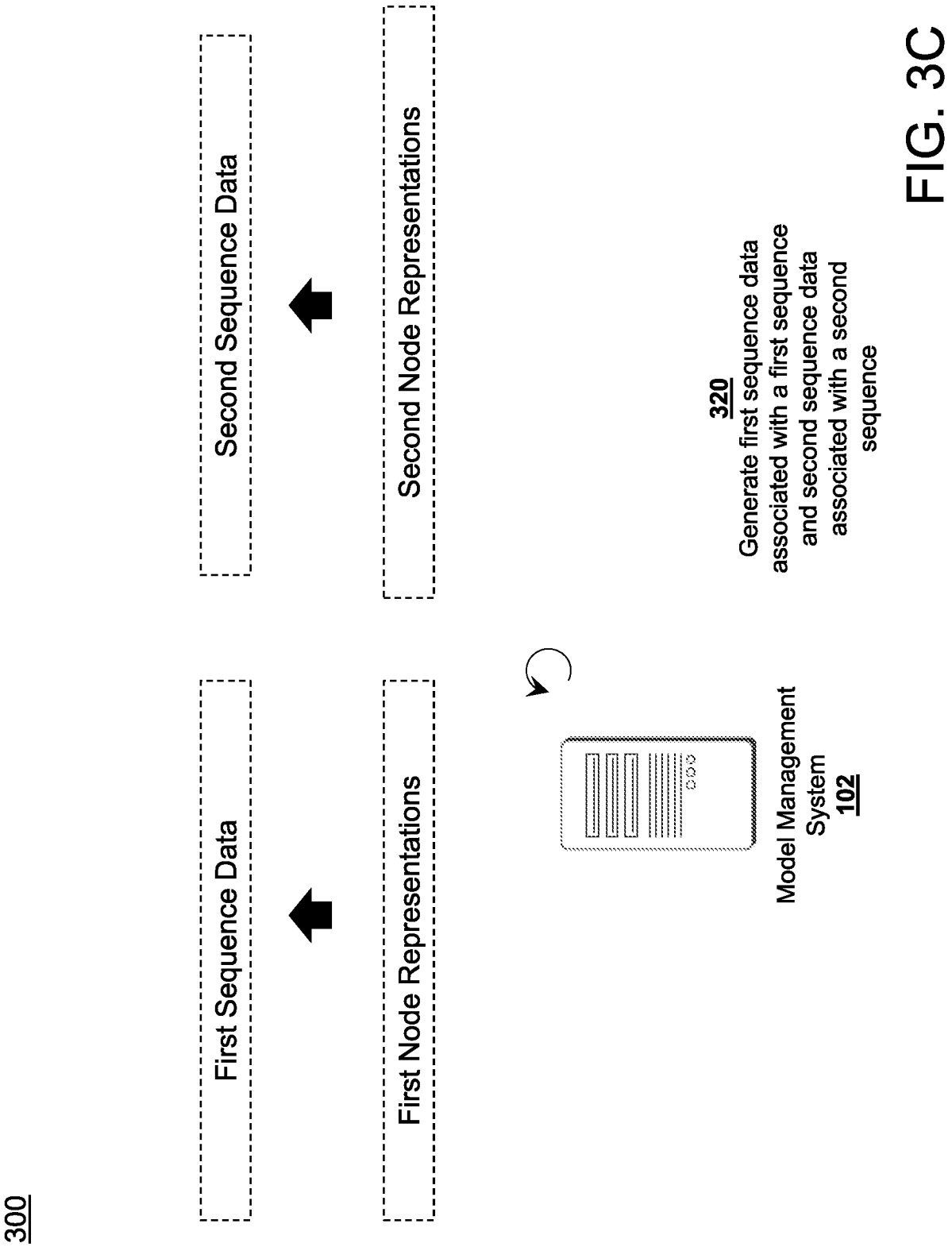
300
First Sequence Data
First Node Representations
Second Sequence Data
Second Node Representations
320
Generate first sequence data associated with a first sequence and second sequence data associated with a second sequence
Model Management System
102

FIG. 3C

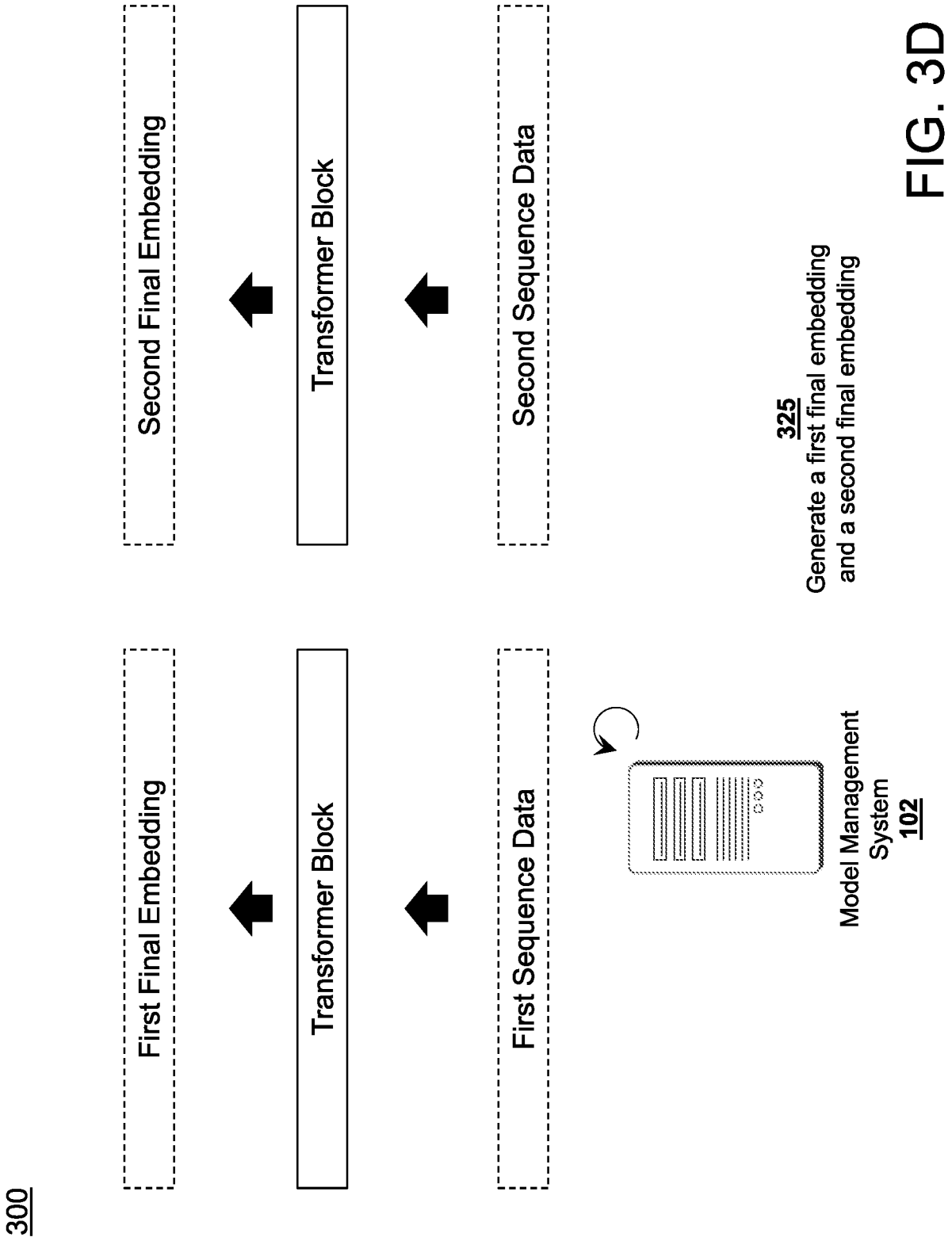
300
First Final Embedding
Transformer Block
First Sequence Data
Second Final Embedding
Transformer Block
Second Sequence Data
Model Management System
102
325
Generate a first final embedding and a second final embedding

FIG. 3D

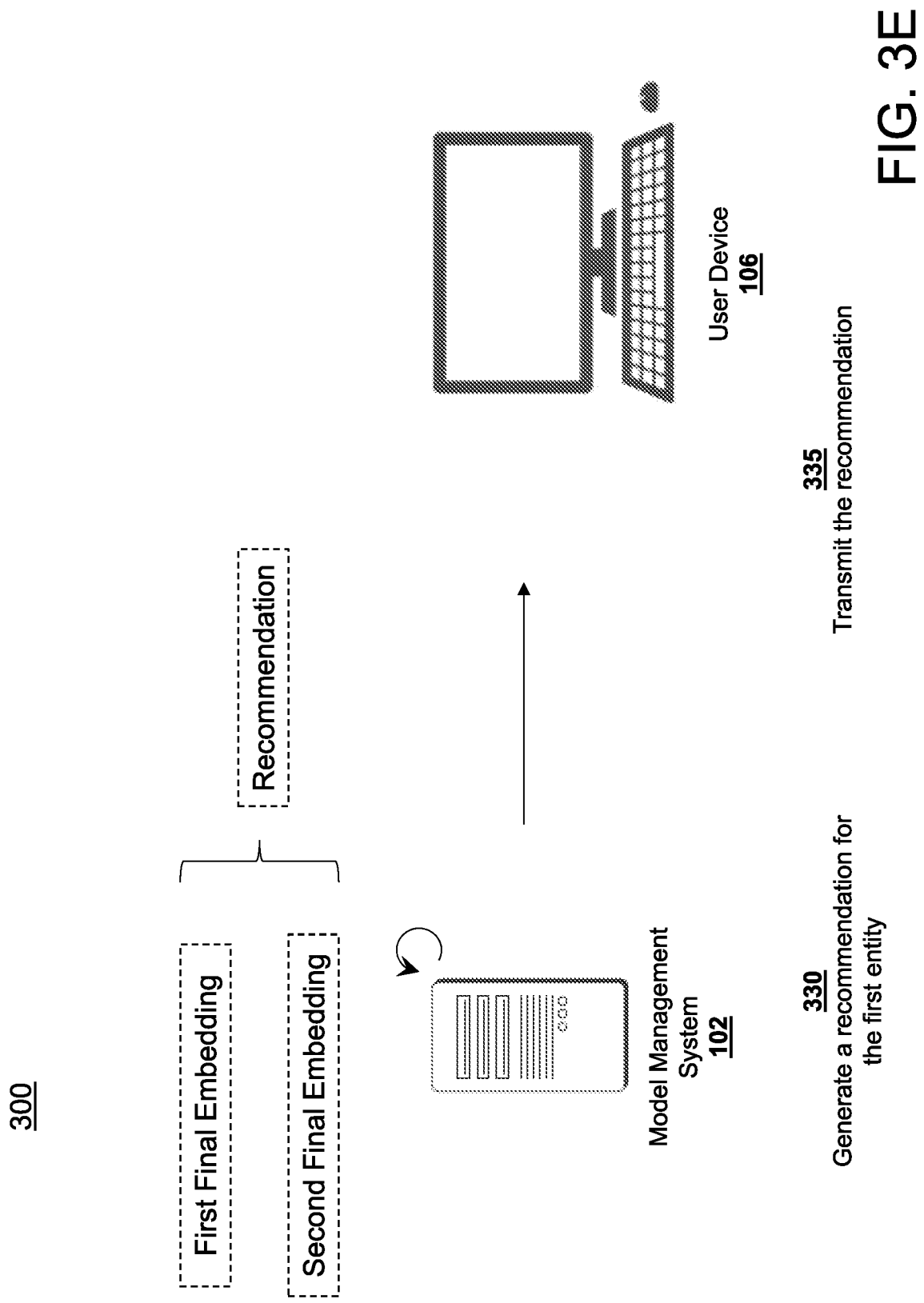
300
First Final Embedding
Second Final Embedding
Recommendation
Model Management System 102
330
Generate a recommendation for the first entity
335
Transmit the recommendation
User Device 106

FIG. 3E

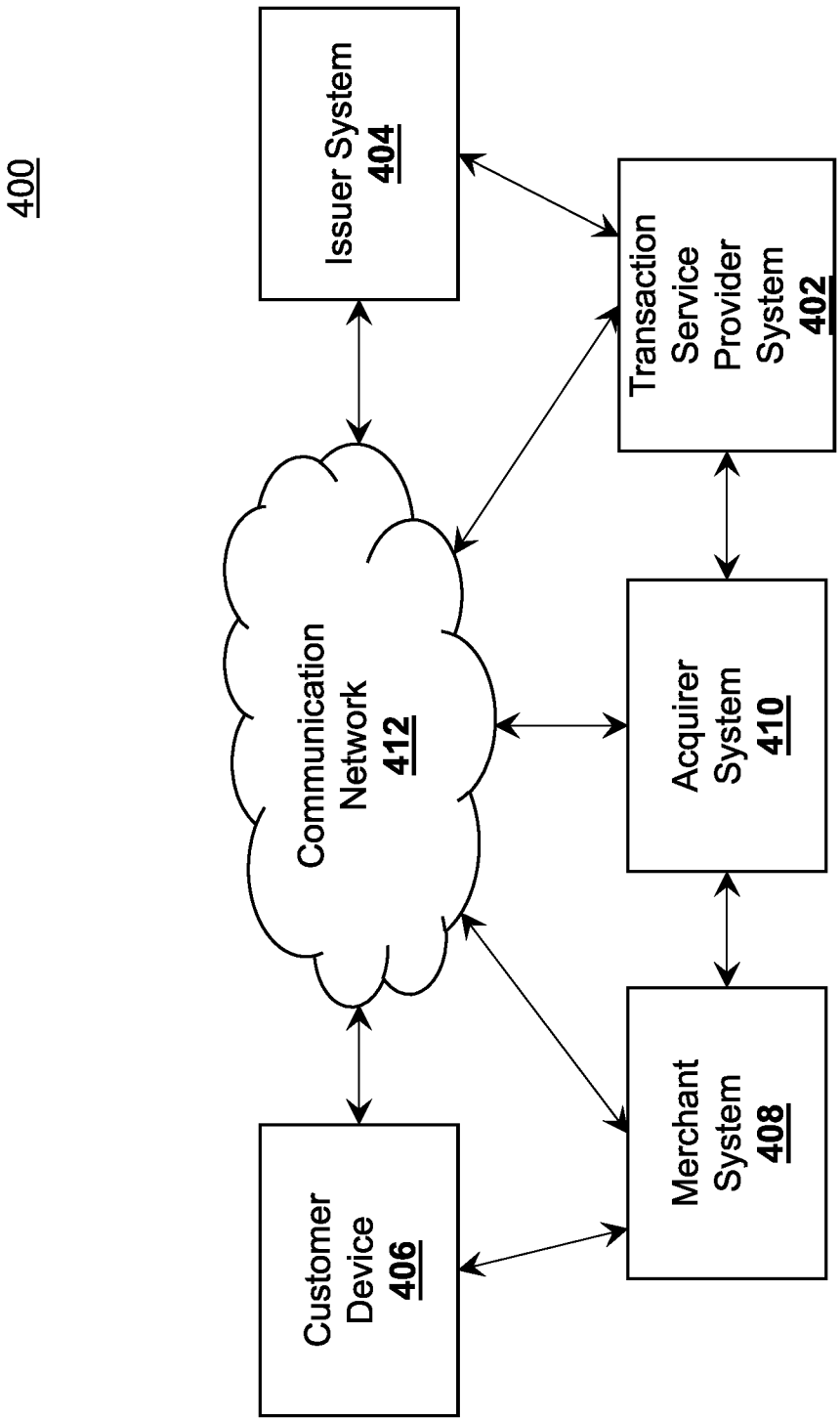
400
Issuer System
404
Transaction
Service
Provider
System
402
Communication
Network
412
Acquirer
System
410
Merchant
System
408
Customer
Device
406

FIG. 4

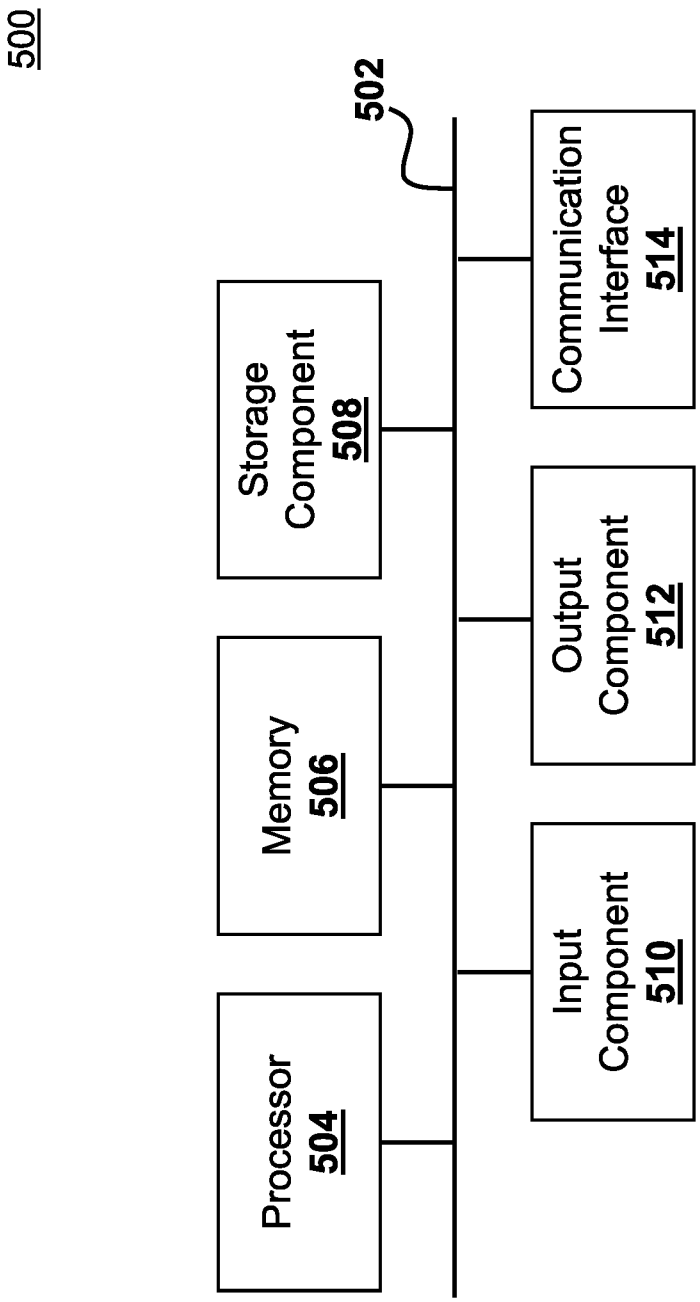
500
502
Processor
504
Memory
506
Storage
Component
508
Input
Component
510
Output
Component
512
Communication
Interface
514

FIG. 5