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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 December 2024 (12.12.2024)

WIPO | PCT

(10) International Publication Number
WO 2024/253898 A1

(51) International Patent Classification:
*G16H 50/20* (2018.01) *G06N 3/045* (2023.01) *A61B 5/00* (2006.01)

(21) International Application Number: PCT/US2024/031274

(22) International Filing Date: 28 May 2024 (28.05.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/506,245 05 June 2023 (05.06.2023) US

(71) Applicant: **GENZYME CORPORATION** [US/US]; 450 Water Street, Cambridge, Massachusetts 02141 (US).

(72) Inventors: **MARTINEZ IPARRAGUIRRE, Mikel**; c/o Genzyme Corporation, 450 Water Street, Cambridge, Massachusetts 02141 (US). **NJOUM, Haneen**; c/o Genzyme Corporation, 450 Water Street, Cambridge, Massachusetts 02141 (US). **SARKER, Hillol**; c/o Genzyme Corporation, 450 Water Street, Cambridge, Massachusetts 02141 (US). **ABOU ZEID, Elias**; c/o Genzyme Corporation, 450 Water Street, Cambridge, Massachusetts 02141 (US).

(74) Agent: **GRAF, Iulia** et al.; Fish & Richardson P.C., P.O. Box 1022, Minneapolis, Minnesota 55440-1022 (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, 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).

(54) **Title:** SLEEP-WAKE CLASSIFICATION USING MACHINE LEARNING

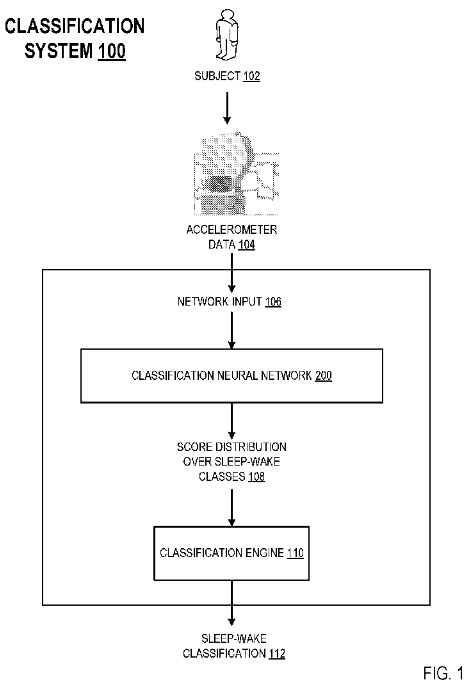


FIG. 1

(57) **Abstract:** Methods, systems, and apparatus, including computer programs encoded on a computer storage medium, for performing a sleep-wake classification task using a sleep-wake classification neural network. In one aspect, a method comprises: receiving accelerometer data characterizing a subject; processing the accelerometer data using a sleep-wake classification neural network, comprising: processing the accelerometer data using an encoder subnetwork of the sleep-wake classification neural network to generate an embedding of the accelerometer data; and processing the embedding of the accelerometer data using a projection subnetwork of the sleep-wake classification neural network to generate a score distribution over a set of sleep-wake classes; classifying a sleep-wake state of the subject based on the respective score for each class in the set of sleep-wake classes; wherein the encoder subnetwork of the sleep-wake classification neural network has been pre-trained to perform an auxiliary task, wherein the auxiliary task is different than the sleep-wake classification task.

WO 2024/253898 A1

[Continued on next page]

**Published:**
— *with international search report (Art. 21(3))*

# SLEEP-WAKE CLASSIFICATION USING MACHINE LEARNING

## TECHNICAL FIELD

**[0001]** This specification relates to sleep-wake classification using machine learning.

## BACKGROUND

**[0002]** Machine learning models receive an input and generate an output, e.g., a predicted output, based on the received input. Some machine learning models are parametric models and generate the output based on the received input and on values of the parameters of the model.

**[0003]** Some machine learning models are deep models that employ multiple layers of models to generate an output for a received input. For example, a deep neural network is a deep machine learning model that includes an output layer and one or more hidden layers that each apply a non-linear transformation to a received input to generate an output.

**[0004]** Sleep is a complex process that can be divided two broad categories: non-rapid eye movement (NREM) sleep and rapid eye movement (REM) sleep. NREM sleep is further divided into stages one through three. Stage one is the lightest stage of sleep, in which the body begins to relax and the breathing and heart rate slow down. Hypnic jerks are common during the transition into stage one. Further, during stage one, alpha wave neural oscillations can decrease while theta wave neural oscillations can increase. Stage two is a deeper level of sleep in which the body begins to prepare for a deeper level of rest. Electroencephalogram (EEG) recordings during stage two characteristically show short bursts of high frequency brain activity ("sleep spindles") and K-complexes. Stage three is the deepest stage of sleep, in which the body experiences a restorative and rejuvenating sleep, repairs itself, builds new tissues, and stores new memories. REM sleep is the final stage of sleep and is characterized by increased brain activity and rapid movement of the eyes. REM sleep is essential for memory consolidation and learning.

## SUMMARY

**[0005]** This specification describes a classification system implemented as computer programs on one or more computers in one or more locations that can process accelerometer data characterizing motion of a subject to generate a sleep-wake classification that characterizes a state of sleep or wakefulness of the subject.

**[0006]** Throughout this specification, a "subject" refers to a human subject, i.e., a person.

**[0007]** Throughout this specification, a "subnetwork" of a neural network refers to a portion of the neural network.

**[0008]** Throughout this specification, an "embedding" of an entity (e.g., of an accelerometer signal) refers to a representation of the entity in a latent space that is generated by a neural network (or a subnetwork of a neural network). An embedding can be represented as an ordered collection of numerical values, e.g., a vector, matrix, or other tensor of numerical values.

**[0009]** Throughout this specification, a "block" (e.g., a "convolutional block" or a "recurrent block") refers to a group of neural network layers in a neural network.

**[0010]** Throughout this specification, the terms "accelerometer data" and "accelerometer signal" are used interchangeably.

**[0011]** Particular embodiments of the subject matter described in this specification can be implemented so as to realize one or more of the following advantages.

**[0012]** The classification system described in this specification can process accelerometer data characterizing motion of a subject to perform a sleep-wake classification task, that is, to classify a state of sleep or wakefulness of the subject. The classification neural network can directly process raw accelerometer data to generate sleep-wake classifications without the need for manual or heuristic feature engineering. More specifically, the classification neural network can be trained, by machine learning training techniques, to identify and extract relevant feature representations from raw accelerometer data in order to accurately classify the sleep-wake state of subjects.

**[0013]** Obtaining labeled training data for training the classification neural network to perform the sleep-wake classification task is time consuming and expensive. ("Labeled" training data refers to accelerometer signals that have been labeled, e.g., by a physician, with a corresponding sleep-wake state). For instance, obtaining labeled training examples for a subject can require performing a sleep study on the subject, where the subject sleeps in a medical facility and is monitored using sensors such as accelerometer sensors, encephalography (EEG) sensors, electroocoulogram (EOG) sensors, electromyography (EMG) sensors, and electrocardiogram (ECG) sensors. A physician (or other expert) can then review the array of multi-modal data generated during the sleep study and laboriously label small segments of the accelerometer data with the sleep-wake state of the subject. Because of the difficulty of obtaining labeled training data, relatively small amounts of training data are available for the training the classification neural network to perform the sleep-wake classification task. However, the performance of machine learning models such as the classification neural network can heavily depend on the availability of large amounts of training data, and a lack of training data can limit the performance of such machine learning models.

**[0014]** To address this issue, the classification system described in this specification can leverage large amounts of readily available unlabeled accelerometer data. In particular, the classification system can train the classification neural network by a two stage process that involves first pre-training an encoder subnetwork of the classification neural network on one or more auxiliary tasks, and then fine-tuning the classification neural network to perform sleep-wake classification. The system can pre-train the encoder subnetwork using the unlabeled accelerometer data and in this way can encourage the encoder subnetwork to generate rich and informative features characterizing the structure of accelerometer data. Pre-training the encoder subnetwork using unlabeled accelerometer data can dramatically reduce the amount of labeled training data required to fine-tune the classification neural network to perform the sleep-wake classification task. Pre-training the encoder subnetwork can thus enable the classification neural network to be trained to achieve an acceptable prediction accuracy on the sleep-wake classification task even when only a relatively small amount of labeled sleep-training training data is available. Further, pre-training the encoder subnetwork can reduce consumption of computational resources, e.g., memory and computing power, during fine-tuning of the classification neural network.

**[0015]** The details of one or more embodiments of the subject matter of this specification are set forth in the accompanying drawings and the description below. Other features, aspects, and advantages of the subject matter will become apparent from the description, the drawings, and the claims.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0016]** FIG. 1 shows an example classification system.

**[0017]** FIG. 2 shows an example architecture of a classification neural network.

**[0018]** FIG. 3A shows an example architecture of a classification neural network that is implemented using one or more residual blocks.

**[0019]** FIG. 3B shows an example architecture of a residual block included in a classification neural network.

**[0020]** FIG. 4 is a flow diagram of an example process for performing a sleep-wake classification task using a sleep-wake classification neural network.

**[0021]** FIG. 5 shows an example training system.

**[0022]** FIG. 6A is a flow diagram of an example process for pre-training the encoder subnetwork of the classification neural network to perform a contrastive embedding task.

**[0023]** FIG. 6B provides an illustration of pre-training the encoder subnetwork to perform a contrastive embedding task.

**[0024]** FIG. 7 is a flow diagram of an example process for pre-training the encoder subnetwork of the classification neural network to perform a masked reconstruction task.

**[0025]** FIG. 8 is a flow diagram of an example process for pre-training the encoder subnetwork of the classification neural network to perform a noisy reconstruction task.

**[0026]** FIG. 9 is a flow diagram of an example process for pre-training the encoder subnetwork of the classification neural network to perform a supervised auxiliary task.

**[0027]** FIG. 10 is a flow diagram of an example process for training the classification neural network to perform a sleep-wake classification task.

**[0028]** FIG. 11 is a flow diagram of an example process for training the classification neural network to perform a medical condition classification task.

**[0029]** FIG. 12A shows experimental results that illustrate the effects of pre-training the encoder neural network of the classification neural network.

**[0030]** FIG. 12B shows experimental results that illustrate the sleep-wake classification accuracy of the system described in this specification in comparison to two other sleep-wake classification systems.

**[0031]** Like reference numbers and designations in the various drawings indicate like elements.

## DETAILED DESCRIPTION

**[0032]** FIG. 1 shows an example classification system 100. The classification system 100 is an example of a system implemented as computer programs on one or more computers in one or more locations in which the systems, components, and techniques described below are implemented.

**[0033]** The classification system 100 is configured to receive accelerometer data 104 characterizing movement of a subject 102. The classification system 100 processes the accelerometer data 104 to generate a sleep-wake classification 112 that characterizes a state of sleep or wakefulness of the subject 102.

**[0034]** The accelerometer data 104 can characterize the movement of the subject 102 over an interval of time, e.g., a 30 second, 60 second, or 90 second interval of time. The accelerometer data 104 can be generated by an accelerometer device located on or in proximity to the subject

102. For instance, the accelerometer data 104 can be generated by an accelerometer in a wearable device being worn by the subject 102, e.g., a wearable device worn on the wrist of the subject.

**[0035]** The classification system 100 can represent the accelerometer data 104 in any of a variety of formats. For instance, the classification system 100 can represent the accelerometer data 104 by one or more one-dimensional (1D) temporal signals, e.g., by respective 1D temporal signals that characterizes acceleration in each of the x-direction, the y-direction, and the z-direction (where the x-, y-, and z- directions are defined with reference to an appropriate frame of reference). As another example, the classification system 100 can represent the accelerometer data 104 by a two-dimensional (2D) spectrogram in the time-frequency domain, i.e., that includes one dimension representing time and a second dimension representing frequency.

**[0036]** Optionally, the classification system 100 can receive additional inputs, i.e., in addition to the accelerometer data 104. The additional inputs can include any appropriate data characterizing the subject 102 or the environment in the vicinity of the subject 102. The classification system 100 can process the additional inputs as part of generating the sleep-wake classification 112. The additional inputs can be derived from any appropriate modalities and can provide the classification system 100 with additional sources of information for increasing the accuracy of the sleep-wake classification 112. A few examples of additional inputs to the classification system 100 are described next.

**[0037]** In some implementations, the classification system 100 additionally receives cardiovascular data characterizing the state of the cardiovascular system of the subject 102. The cardiovascular data can be generated using one or more cardiovascular sensors (e.g., a heart rate monitor or a pulse oximeter) located on or in proximity to the subject, e.g., in a wearable device being worn by the subject 102. In some implementations, the cardiovascular data can include data defining a heart rate of the subject, or a variability (e.g., variance) in the heart rate of the subject, or a blood oxygen saturation level of the subject, over an interval of time. In some cases, the cardiovascular data can include time series data, e.g., that defines the value of a cardiovascular parameter at each time point in a sequence of time points over a time interval. For instance, the cardiovascular data can include one or more audio waveforms that characterize heart sounds generated by the heart of the subject. The cardiovascular data can be captured over the same time interval as the accelerometer data 104.

**[0038]** In some implementations, the classification system 100 additionally receives data characterizing ambient light in the vicinity of the subject 102. The term "ambient light" can refer to any light, e.g., natural light (e.g., sunlight or moonlight), or artificial light (e.g., generated by one or more lighting devices). The ambient light data can be generated using one or more light sensors located in the vicinity of the subject. The ambient light data can characterize any appropriate features of the light in the vicinity of the subject, e.g., the intensity of the light, the color of the light, the variation in the light, etc. The ambient light data can include aggregated data, e.g., one or more statistics that summarize the light in the vicinity of the subject over a time interval, or time series data, e.g., that define respective values of one or more light parameters at each time point in a sequence of time points over a time interval. The ambient light data can be captured over the same time interval as the accelerometer data 104.

**[0039]** In some implementations, the classification system 100 additionally receives audio data characterizing sound in the vicinity of the subject 102. The audio data can be generated using one or more sound sensors located in the vicinity of the subject 102. The audio data can include aggregated data (e.g., one or more statistics that summarize the intensity or variation of the sound in the vicinity of the subject over a time interval) or time series data (e.g., one or more audio waveforms generated by one or more sound sensors located in the vicinity of the subject 102 over a time interval). The audio data can be captured over the same time interval as the accelerometer data 104.

**[0040]** In some implementations, the classification system 100 additionally receives data time data characterizing a current time of day when the accelerometer data 104 is captured. The classification system 100 can represent the time data in any appropriate way, e.g., by a one-hot embedding vector. The one-hot embedding vector can include a respective entry for each hour in the day, where the entry corresponding to the current hour has value one (or some other predefined value) and the entries corresponding to hours other than the current hour have value zero (or some other predefined value).

**[0041]** In some implementations, the classification system 100 additionally receives electroencephalography (EEG) data that characterizes electrical activity in the brain of the subject 102. The EEG data can be generated using an electroencephalogram device that measures electrical activity in the brain of the subject using small metal disks (electrodes) attached to the scalp of the subject. The EEG data can include aggregated data (e.g., one or more statistics that summarize the intensity or variation of the electrical activity in the brain over a time interval) or time series data (e.g., a respective electrical activity waveform generated by each of one or more EEG electrodes attached to the scalp over the subject over a time interval). The EEG data can be captured over the same time interval as the accelerometer data 104.

**[0042]** In some implementations, the classification system 100 additionally receives video data that includes a video that shows part of the subject (e.g., the face of the subject or the torso of the subject) or all of the subject (e.g., the entire body of the subject) over an interval of time. The video data can be captured by a video recording device, e.g., a webcam, or a video recording device in a smartphone. The video data can be represented as a sequence of video frames captured at any appropriate sampling frequency, e.g., 24 frames-per-second. The video data can be captured over the same time interval as the accelerometer data 104.

**[0043]** The classification system 100 can process the accelerometer data 104, and any additional inputs characterizing the subject 102, using a classification neural network 200 and a classification engine 110, which are each described next. (The classification neural network 200 is also referred to throughout this specification as the "sleep-wake" classification neural network).

**[0044]** The classification neural network 200 is configured to receive a network input 106 that includes the accelerometer data 104, and optionally, any additional inputs characterizing the subject or the environment of the subject, e.g., cardiovascular data, ambient light data, audio data, time data, EEG data, video data, etc. The classification neural network 200 processes the network input 106, in accordance with values of a set of classification neural network parameters, to generate a score distribution over a set of sleep-wake classes.

**[0045]** The score distribution 112 over the set of sleep-wake classes can define a respective score for each sleep-wake class in the set of sleep-wake classes. Each sleep-wake class can correspond to a respective state of sleep or wakefulness. The score for a sleep-wake class can define a likelihood that the subject is in the corresponding state of sleep or wakefulness. The set of sleep-wake classes can include any appropriate number of sleep-wake classes, e.g., two sleep-wake classes, three sleep-wake classes, four sleep-wake classes, or five sleep-wake classes. In particular, the set of sleep-wake classes can include: (i) a "wake" class indicating that the subject is awake, and (ii) one or more "sleep" classes indicating that the subject is in a respective stage of sleep.

**[0046]** In some implementations, the set of sleep-wake classes can include respective sleep classes corresponding to "non-REM sleep" (indicating that the subject is in a non-REM sleep stage) and "REM sleep" (indicating that the subject is in the REM sleep stage). In some implementations, the set of sleep-wake classes can include respective sleep classes corresponding to "stage 1 non-REM sleep," "stage 2 non-REM sleep," "stage 3 non-REM sleep," and "REM sleep." In some implementations, the set of sleep-wake classes can include only two sleep-wake classes: a "wake" class and a "sleep" class, where the "sleep" class indicates that the subject is in any stage of sleep.

**[0047]** In implementations where the set of sleep-wake classes includes only two classes, e.g., a "wake" class and a "sleep" class, the classification neural network 200 can be configured to generate an output that includes a score for only one of the two classes. For instance, the classification neural network 200 can be configured to generate a score that defines a likelihood that the subject is included in the "wake" class, or a score that defines a likelihood that the subject is included in the "sleep" class. A score for one class (from the set of two classes) can define a score for the other class, for instance, based on the requirement that the two scores sum to one (or some other predefined value). Thus a score for one class can implicitly define a score distribution over the set of two classes.

**[0048]** The classification neural network 200 can have any appropriate neural network architecture that enables the classification neural network 200 to perform its described functions, e.g., generating a score distribution over a set of sleep-wake classes. In particular, the classification neural network can include any appropriate neural network layers (e.g., convolutional layers, attention layers, fully connected layers, recurrent layers, etc.) in any appropriate number (e.g., 5 layers, 10 layers, or 50 layers) and connected in any appropriate configuration (e.g., as a linear sequence of layers). An example architecture of the classification neural network is described in more detail with reference to FIG. 2.

**[0049]** The classification system 100 can use a training system to train the classification neural network 200 to perform the task of sleep-wake classification. Optionally, the training system can pre-train portions of the classification neural network 200 to perform one or more unsupervised or supervised auxiliary tasks prior to training the classification neural network 200 to perform the sleep-wake classification task. An example of a training system for training the classification neural network is described in more detail with reference to FIG. 5.

**[0050]** The classification engine 110 is configured to process a score distribution 112 over the set of sleep-wake classes to generate a sleep-wake classification 112. The sleep-wake classification engine 110 can, for instance, generate a sleep-wake classification that classifies the subject as being included in the sleep-wake class associated with the highest score under the score distribution 112 over the set of sleep-wake classes.

**[0051]** Optionally, the classification engine 110 can generate a confidence score that characterizes a confidence of the classification system 100 in the sleep-wake classification 112 generated for the subject 102. The classification engine 110 can generate a confidence score for a sleep-wake classification 112 in any appropriate way. For instance, the classification engine 110 can generate a confidence score by computing an entropy of the score distribution over the set of sleep-wake classes 108. In this example, a higher entropy of the score distribution over the set of sleep-wake classes 108 can reflect a higher uncertainty of the classification system 100 in the sleep-wake classification 112.

**[0052]** The classification system 100 can use sleep-wake classifications 112 generated for the subject 102 in any of a variety of ways. A few example applications of sleep-wake classifications 112 are described next.

**[0053]** In some implementations, the classification system 100 receives accelerometer data 104 (and optionally other inputs, such as video data or audio data) for each time interval in a sequence of time intervals. The classification system 100 can process the accelerometer data 104 for each time interval to generate a corresponding sleep-wake classification 112. The classification system 100 can thus generate a sequence of sleep-wake classifications 112, where each sleep-wake classification 112 corresponds to a respective time interval and characterizes the sleep-wake state of the subject 102 over the time interval. The sequence of sleep-wake classifications can provide granular and real-time monitoring of the sleep-wake state of the subject 102, without requiring the subject 102 to undergo a sleep study.

**[0054]** In some implementations, the classification system 100 can process a sequence of sleep-wake classifications (as described above) to predict a duration of time, over a time window (e.g., a 24 hour time window), that the subject 102 was in a particular sleep-wake state. For instance, the classification system 100 can predict a duration of time that the subject 102 was in a particular sleep-wake class over a time window based on a combination (e.g., a product) of: (i) a number of time intervals (over the time window) where the classification system 100 classified the subject as being in the sleep-wake class, and (ii) a duration of the time intervals (e.g., 90 seconds). As another example, the classification system 100 can predict a duration of time that the subject 102 was in a specified proper subset of the set of sleep-wake classes over a time window based on a combination of: (i) a number of time intervals (over the time window) where the classification system 100 classified the subject as being in a sleep-wake class include in the proper subset of the sleep-wake classes, and (ii) a duration of the time intervals (e.g., 90 seconds). Thus, for instance, the classification system can predict, e.g., a duration of time that the subject 102 was asleep based on the number of time intervals over which the subject was classified as being in stage 1 non-REM sleep, stage 2 non-REM sleep, stage 3 non-REM sleep, or REM sleep.

**[0055]** The classification system 100 can generate a notification that indicates the duration of the time that the subject is in a particular sleep-wake class (or in a subset of sleep-wake classes) over a time window, and can provide the notification, e.g., to the subject 102 or to a healthcare provider of the subject 102. For instance, the classification system 100 can transmit the notification over a data communications network, e.g., as an email or a text message. Further, the classification system 100 can automatically store sleep-wake classifications 112 generated for the subject 102 (or data derived from the sleep-wake classifications, as described above) in an electronic medical record of the subject 102. For instance, the classification system 100 can interface with a database storing electronic medical records of the subject 102, e.g., by way of an application programming interface (API), and store the sleep-wake classifications (or data derived from the sleep-wake classifications) in an appropriate field in an electronic medical record of the subject 102.

**[0056]** In some implementations, the classification system 100 can trigger one or more actions based on determining that the subject 102 has been included in a particular sleep-wake class (or in a subset of sleep-wake classes) for a threshold duration of time. For instance, the classification system 100 can trigger an alarm sound or turn on a light to cause the subject 102 to transition into a state of wakefulness in response to determining that the subject 102 has been asleep for at least a threshold duration of time, e.g., eight hours. As another example, the classification system 100 can trigger a notification (e.g., a text message) indicating that the subject 102 should consider sleeping in response to determining that the subject 102 has been awake for at least a threshold duration of time, e.g., 20 hours.

**[0057]** In some implementations, the classification system 100 can generate an alarm in response to generating one or more sleep-wake classifications 112 indicating that the user has transitioned into a state of sleep. The alarm can be configured to cause the user to transition back to a state of wakefulness. In a particular application, the classification system 100 can be used to maintain wakefulness of a subject 102 that is in an environment or engaged in an activity where being asleep is not desirable (e.g., driving a vehicle). The alarm can be, e.g., a sound alarm, or a visual alarm (e.g., a flashing light), or a kinetic alarm (e.g., a vibration of a wearable device of the subject 102).

**[0058]** Optionally, in combination with or as an alternative to generating sleep-wake classifications, the classification system 100 can generate medical condition classifications. More specifically, the classification neural network 200 can be configured to process a network input 106 that characterizes the subject 102 over a time interval to generate a score that defines a likelihood that the subject displays symptoms of a medical condition during the time interval. The network input 106 can include accelerometer data 104, and optionally other types of data (e.g., video data, audio data, etc.). The medical condition can be, e.g., restless leg syndrome or seizures. The classification system 100 can classify the subject 102 as exhibiting a symptom of the medical condition over a time interval, e.g., if the likelihood that the subject exhibits the symptom over the time interval (as defined by the output of the classification neural network 200) satisfies (e.g., exceeds) a threshold.

**[0059]** The classification system 100 can generate a sequence of medical condition classifications, including a respective medical condition classification for each time interval in a sequence of time intervals. The classification system 100 can process the sequence of medical condition classifications to generate a medical diagnosis. For instance, the classification system 100 can generate a medical diagnosis indicating that the subject 102 is predicted to have a medical condition in response to generating at least a threshold number of medical condition classifications indicating that the subject has displayed symptoms of the medical condition. After generating a medical diagnosis, the classification system 100 can generate a notification that indicates the diagnosis, and transmit the notification, e.g., to the subject 102 or to a healthcare provider of the subject 102.

**[0060]** FIG. 2 shows an example architecture of a classification neural network 200, e.g., that is included in the classification system 100 described with reference to FIG. 1. The classification neural network 200 is configured to receive a network input 106 that includes accelerometer data characterizing movement of a subject. The network input 106 can include additional data characterizing the subject or the environment of the subject, e.g., cardiovascular data, video data, audio data, time data, etc., as described above with reference to FIG. 1. The classification neural network 200 processes the network input 106 to generate a score distribution over a set of sleep-wake classes 112.

**[0061]** The classification neural network 200 includes an encoder subnetwork 202 and a projection subnetwork 206, which are each described in more detail next.

**[0062]** The encoder subnetwork 202 is configured to process the network input 106 to generate an embedding 204 of the network input 106 in a latent space. The encoder subnetwork 202 can have any appropriate neural network architecture that enables the encoder subnetwork 202 to generate an embedding 204 of a network input 106. In particular, the encoder subnetwork can include any appropriate types of neural network layers (e.g., convolutional layers, recurrent layers, attention layers, fully connected layers, etc.) in any number (e.g., 5 layers, 10 layers, or 50 layers) and in any appropriate configuration (e.g., as a linear sequence of layers). A specific example of an architecture of an encoder subnetwork 202 is described in more detail with reference to FIG. 3A.

[0063] The network input 106 can include data from multiple modalities, e.g., accelerometer data, video data, sound data, time data, cardiovascular data, etc. The architecture of the encoder subnetwork 202 can be configured in any of a variety of possible ways to enable the encoder subnetwork 202 to process multi-modal data. For instance, the encoder subnetwork 202 can include a respective sequence of encoder neural network layers corresponding to each modality. For each modality, the encoder subnetwork 202 can process data derived from that modality using the corresponding sequence of encoder neural network layers to generate a modality-specific embedding of the data. The encoder subnetwork 202 can combine the modality-specific embedding of each modality included in the network input 106 to generate the embedding 204 of the network input 106. The encoder subnetwork 202 can combine the modality-specific embeddings, e.g., by concatenating the modality-specific embeddings, or by pooling (e.g., averaging, summing, or max pooling) the modality-specific embeddings.

[0064] The projection subnetwork 206 is configured to process the embedding 204 of the network input 106 to generate a score distribution over the set of sleep-wake classes 108. The projection subnetwork 206 can have any appropriate neural network architecture that enables the projection subnetwork 206 to generate a score distribution over a set of sleep-wake classes. In particular, the projection subnetwork can include any appropriate types of neural network layers (e.g., convolutional layers, recurrent layers, attention layers, fully connected layers, etc.) in any number (e.g., 5 layers, 10 layers, or 50 layers) and in any appropriate configuration (e.g., as a linear sequence of layers). In a specific example, the projection subnetwork 206 can include a sequence of dense (fully connected) neural network layers. Another specific example of a projection subnetwork 206 is described with reference to FIG. 3A.

[0065] Optionally, the classification neural network 200 can include a second projection subnetwork that is configured to process the embedding 204 of the network input 106 to generate a score that defines a likelihood that the subject displays symptoms of a medical condition, e.g., restless leg syndrome or seizures.

[0066] FIG. 3A shows an example architecture of a classification neural network that is implemented using one or more residual blocks. The classification neural network can be included in a classification system 100, as described with reference to FIG. 1. The classification neural network 200 is configured to process a network input 106 that includes accelerometer data characterizing a subject (and, optionally, additional data such as cardiovascular data, video data, audio data, time data, etc.) to generate a network output that defines a score distribution over a set of sleep-wake classes 112. The classification neural network includes an encoder subnetwork 202 and a projection subnetwork 206.

[0067] In the example architecture illustrated in FIG. 3A, the classification neural network includes residual blocks (e.g., 302, 308, 314, 320), pooling layers (e.g., 304, 310, 316, 322, 326), dropout layers (e.g., 306, 312, 318, 324), and one or more fully connected layers (e.g., 328).

[0068] A residual block refers to a block (i.e., a group of neural network layers) that is configured to process a block input by one or more neural network layers of the block to generate an intermediate output, and then to generate a block output by combining (e.g., summing) the block input and the intermediate output. That is, a residual block includes a shortcut connection that enables the input to the block to be combined with the output of the block. Including residual blocks in a neural network can improve the stability of the neural network and allow the neural network to learn more efficiently, e.g., by reducing the effect of vanishing gradients during training. An example architecture of a residual block is illustrated with reference to FIG. 3B.

[0069] The architecture of a residual block can be defined in part by a "kernel" parameter, a "dilation" parameter, and a "filter" parameter. The kernel parameter can define, e.g., the number of weights included in filters of convolutional neural network layers of the residual block. The "dilation" parameter can define a spacing between weights included in filters of convolutional neural network layers of the residual block. The "filter" parameter can define a number of convolutional filters in the convolutional neural network layers of the residual block.

[0070] The pooling layers are configured to receive a layer input that includes a set of embeddings, and to combine the embeddings included in the layer input by a pooling operation to generate a layer output that includes fewer embeddings. The pooling operation can be, e.g., a maximum operation or an averaging operation. The architecture of a pooling layer can be defined in part by a "pool" parameter and a "stride" parameter. The "pool" parameter can define a number of embeddings that the pooling operation is configured to combine into a single embedding. The "stride" parameter can define a resolution of the pooling operation.

[0071] The dropout layers operate by randomly setting some input units to zero with a certain probability during training of the neural network, thus reducing the risk of overfitting. The architecture of a dropout layer can be defined in part by a "probability" parameter that defines a likelihood that each unit of the input to the dropout layer will be set to zero during training.

[0072] FIG. 3B shows an example architecture of a residual block included in a classification neural network, e.g., the classification neural network illustrated in FIG. 3A. The residual block can be configured to process a set of input embeddings derived from accelerometer data characterizing motion of a subject, in accordance with values of a set of residual block parameters, to generate a set of output embeddings. The residual block includes a skip (shortcut) connection 330, one-dimensional (1-D) convolutional layers (e.g., 332, 336, 340), batch normalization (BN) layers (e.g., 334, 338, 342), and rectified linear unit (ReLU) layers (e.g., 334, 338, 344).

**[0073]** FIG. 4 is a flow diagram of an example process 400 for performing a sleep-wake classification task using a sleep-wake classification neural network. For convenience, the process 400 will be described as being performed by a system of one or more computers located in one or more locations. For example, a classification system, e.g., the classification system 100 of FIG. 1, appropriately programmed in accordance with this specification, can perform the process 400.

**[0074]** Receive a network input that includes accelerometer data generated by a wearable device of a subject (402). Optionally the network input can include additional data, e.g., one or more of: cardiovascular data (e.g., heart rate data, heart rate variability data, blood oxygen saturation data, etc.), EEG data, ambient light data, audio data, time data, video data, etc.

**[0075]** Process the network input using a sleep-wake classification neural network to generate a network output that defines a score distribution over a set of sleep-wake classes (404). More specifically, the system processes the network input (including the accelerometer data) using an encoder subnetwork of the sleep-wake classification neural network to generate an embedding of the accelerometer data in a latent space. The system then processes the embedding of the accelerometer data using a projection subnetwork of the sleep-wake classification neural network to generate the score distribution over the set of sleep-wake classes.

**[0076]** The set of sleep-wake classes includes: (i) at least one class corresponding to a state of wakefulness, and (ii) at least one class corresponding to a state of sleep. The set of sleep-wake classes can include multiple classes corresponding to respective states of sleep. For instance, the set of sleep-wake classes can include a respective class corresponding to each of: stage 1 non-REM sleep, stage 2 non-REM sleep, stage 3 non-REM sleep, and REM sleep.

**[0077]** The sleep-wake classification neural network can include one or more residual blocks. Each residual block is configured to process a block input by one or more neural network layers of the residual block to generate an intermediate output, and to generate a block output by summing: (i) the block input, and (ii) the intermediate output.

**[0078]** The system can train the sleep-wake classification neural network using machine learning training techniques. In particular, the system can pre-train the encoder subnetwork of the sleep-wake classification neural network to perform an auxiliary task, where the auxiliary task is different than the sleep-wake classification task. After pre-training the encoder subnetwork, the system can train the sleep-wake classification neural network to perform the sleep-wake classification task. An example of a training system for training the sleep-wake classification neural network is described with reference to FIG. 5.

**[0079]** The system classifies a sleep-wake state of the subject based on the respective score for each class in the set of sleep-wake classes (406). For instance, the system can classify the sleep-wake state of the subject into a class associated with a highest score from among the set of sleep-wake classes.

**[0080]** FIG. 5 shows an example training system 500. The training system 500 is an example of a system implemented as computer programs on one or more computers in one or more locations in which the systems, components, and techniques described below are implemented.

**[0081]** The training system 500 is configured to train the classification neural network 200 included in the classification system 100 described with reference to FIG. 1. The classification neural network 200 is configured to process a network input that includes accelerometer data characterizing motion of a subject to generate a score distribution over a set of sleep-wake classes.

**[0082]** The classification neural network 200 includes an encoder subnetwork and a projection subnetwork, as described with reference to FIG. 2. The encoder subnetwork can process a network input, including the accelerometer data, to generate an embedding of the network input. The projection subnetwork can process the embedding of the network input to generate a score distribution over the set of sleep-wake classes.

**[0083]** The training system 500 can train the classification neural network over a sequence of two stages. In the first stage, the training system 500 can pre-train the encoder subnetwork 202 of the classification neural network 200 to perform one or more auxiliary tasks. In the second stage, the training system 500 can train the classification neural network 200 to perform the sleep-wake classification task. Pre-training the encoder subnetwork 202 can provide an effective initialization of the parameter values of the encoder neural network, thus facilitating fine-tuning of the classification neural network to perform sleep-wake classification. In particular, the pre-training can encourage the encoder subnetwork 202 to generate embeddings that encode rich features characterizing the structure of accelerometer data (and, optionally, other types of data, e.g., cardiovascular data, video data, audio data, etc.). Pre-training the encoder subnetwork 202 can enable the classification neural network 200 to achieve a higher prediction accuracy on the sleep-wake classification task while requiring less labeled training data than would otherwise be necessary.

**[0084]** At the first stage of training, the training system 500 can initialize the set of parameters of the encoder subnetwork 202 using an appropriate initialization technique, e.g., random initialization or glorot initialization. The training system 500 can then train the encoder subnetwork 202 to perform an auxiliary task. A few examples of auxiliary tasks are described next.

**[0085]** In some implementations, the training system 500 can pre-train the encoder subnetwork 202 to perform a contrastive embedding task 502. More specifically, the training system 500 can train the encoder subnetwork to generate similar embeddings for "positive" pairs of accelerometer signals, and dissimilar embeddings for "negative" pairs of accelerometer signals. A "pair" of accelerometer signals includes a first accelerometer signal and a second accelerometer signal. A "positive" pair of accelerometer signals can refer to a pair of accelerometer signals where the first accelerometer signal and the second accelerometer signal are both transformed versions of a same underlying accelerometer signal. A "negative" pair of accelerometer signals can refer to a pair of accelerometer signals where the first accelerometer signal is derived from a different underlying accelerometer signal than the second accelerometer signal. An example process for training the encoder subnetwork 202 to perform a contrastive embedding task 502 is described in more detail with reference to FIG. 6A-6B.

**[0086]** In some implementations, the training system 500 can pre-train the encoder subnetwork 202 to perform a masked reconstruction task 504. More specifically, the training system can train the encoder subnetwork to process a masked accelerometer signal to generate an embedding that enables accurate reconstruction of the full (unmasked) accelerometer signal. An example process for training the encoder subnetwork 202 to perform a masked reconstruction task 504 is described in more detail with reference to FIG. 7.

**[0087]** In some implementations, the training system 500 can pre-train the encoder subnetwork 202 to perform a noisy reconstruction task 506. More specifically, the training system can train the encoder subnetwork to process a noised accelerometer signal to generate an embedding that enables accurate reconstruction of the original (de-noised) accelerometer signal. An example process for training the encoder subnetwork 202 to perform a noisy reconstruction task 506 is described in more detail with reference to FIG. 8.

**[0088]** In some implementations, the training system 500 can pre-train the encoder subnetwork 202 to perform one or more supervised auxiliary tasks 508. More specifically, the training system can train the encoder subnetwork to process an accelerometer signal to generate an embedding that enables accurate prediction of one or more features of the accelerometer signal, e.g., a number of steps taken by a subject in a duration of time covered by the accelerometer signal, or an action performed by a subject in a duration of time covered by the accelerometer signal. An example process for training the encoder subnetwork to perform a supervised auxiliary task 508 is described in more detail with reference to FIG. 9.

**[0089]** Optionally, the training system 500 can pre-train the encoder subnetwork to perform multiple auxiliary tasks, i.e., rather than just a single auxiliary task. For instance, the training system 500 can pre-train the encoder subnetwork to perform two auxiliary tasks, or three auxiliary tasks, or four auxiliary tasks.

**[0090]** At the second stage of training, the training system 500 can initialize the set of parameters of the projection subnetwork using an appropriate initialization scheme, e.g., random initialization or glorot initialization. (The set of parameters of the encoder subnetwork 202 can have the values determined during the pre-training of the encoder subnetwork at the first stage of training). The training system 500 can then train the classification neural network 200 to perform a sleep-wake classification task. An example process for training the classification neural network to perform a sleep-wake classification task 510 is described in more detail with reference to FIG. 10.

**[0091]** In some implementations, the training system 500 trains the classification neural network 200 to perform both a sleep-wake classification task and medical condition classification task. In these implementations, the classification neural network 200 can include a first projection subnetwork that generates a score distribution over a set of sleep-wake classes and a second projection subnetwork that generates a score defining a likelihood that the subject has a medical condition (as described with reference to FIG. 1). The training system 500 can exploit synergies that exist between the sleep-wake classification task and the medical condition classification task in order to achieve a higher prediction accuracy on both tasks. More specifically, as part of training the classification neural network to perform the sleep-wake classification task, the training system can backpropagate gradients of a sleep-wake classification objective function through the first projection subnetwork and into the encoder subnetwork of the classification neural network. Similarly, as part of training the classification neural network to perform the medical condition classification task, the training system can backpropagate gradients of a medical condition classification objective function through the second projection subnetwork and into the encoder subnetwork of the classification neural network. The parameter values of the encoder subnetwork can thus be jointly trained using training signals from both the sleep-wake classification task and the medical condition classification task, thereby enabling the encoder subnetwork to learn to exploit synergies and commonalities between the tasks. An example process for training the classification neural network to perform a medical condition classification task is described with reference to FIG. 11.

**[0092]** For convenience, the first stage of training – in particular, pre-training the encoder neural network – is described as being performed before the second stage of training – in particular, fine-tuning the classification neural network to perform sleep-wake classification and/or medical condition classification. However, the two stages of training can overlap, e.g., such that the pre-training of the encoder neural network and the fine-tuning of the classification neural network are both performed at one or more of the same training iterations.

**[0093]** After training the classification neural network 200, the training system can provide the classification neural network 200 for use by the classification system 100, as described with reference to FIG. 1.

**[0094]** FIG. 6A is a flow diagram of an example process 600 for pre-training the encoder subnetwork of the classification neural network to perform a contrastive embedding task. For convenience, the process 600 will be described as being performed by a system of one or more computers located in one or more locations. For example, a training system, e.g., the training system 500 of FIG. 5, appropriately programmed in accordance with this specification, can perform the process 600. The training system can iteratively perform the steps of the process 600 as part of pre-training the encoder subnetwork.

**[0095]** The system obtains a set of "base" accelerometer signals (602).

**[0096]** The system generates one or more positive pairs of accelerometer signals (604). To generate a positive pair of accelerometer signals, the system selects (e.g., randomly samples) a base accelerometer signal from the set of base accelerometer signals. The system generates a first transformed version of the base accelerometer signal, e.g., by randomly sampling a first transformation from a space of transformations, and applying the first transformation to the base accelerometer signal. The system generates a second transformed version of the base accelerometer signal, e.g., by randomly sampling a second transformation from the space of transformations, and applying the second transformation to the base accelerometer signal. The first transformed version and the second transformed version of the base accelerometer signal jointly define the positive pair of accelerometer signals.

**[0097]** The space of transformations can include, e.g., a flipping transformation (that includes flipping the accelerometer signals around the time axis), a reversing transformation (that includes reversing the direction of the time axis for the accelerometer signals), a zoom transformation (that includes cropping and resizing the accelerometer signals), a swapping transformation (that includes swapping the ordering of the accelerometer signals), a noising transformation (that includes adding random noise to the accelerometer signals), a resizing transformation (that includes modifying the amplitude of the accelerometer signals), and an identity transformation (that has no effect on accelerometer signals). In some cases, a positive pair of accelerometer signals can include an accelerometer signal that is identical to a base accelerometer signal, e.g., in situations where the system selects the identity transformation to apply to the base accelerometer signal.

**[0098]** The system generates one or more positive pairs of embeddings (606). In particular, for each positive pair of accelerometer signals, the system processes the first accelerometer signal using the encoder subnetwork to generate an embedding of the first accelerometer signal, and the system processes the second accelerometer signal to generate an embedding of the second accelerometer signal. The embedding of the first accelerometer signal and the embedding of the second accelerometer signal jointly define a positive pair of embeddings.

**[0099]** In some implementations, the system uses both the encoder subnetwork and another neural network, referred to for convenience, as an embedding neural network, to generate an embedding of an accelerometer signal during pre-training of the encoder subnetwork. In particular, to generate an embedding of an accelerometer signal, the system can process the accelerometer signal using the encoder subnetwork to generate a first embedding, and then process the first embedding using the embedding neural network to generate a second embedding. The system can define the embedding generated by the embedding neural network as the embedding of the accelerometer signal during the pre-training of the encoder sub-network.

**[0100]** The system generates one or more negative pairs of accelerometer signals (608). To generate a negative pair of accelerometer signals, the system selects (e.g., randomly samples) a pair of base accelerometer signals, including a first base accelerometer signal and a second, different base accelerometer signal. The system generates a transformed version of the first base accelerometer signal, e.g., by randomly sampling a transformation from the space of transformations, and applying the transformation to the first base accelerometer signal. The system generates a transformed version of the second base accelerometer signal, e.g., by randomly sampling a transformation from the space of transformations, and applying the transformation to the second base accelerometer signal. The transformed versions of the first and second base accelerometer signals jointly define the negative pair of accelerometer signals. The system may select the identity transformation for either the first or second base accelerometer signal, such that the negative pair of accelerometer signals includes base accelerometer signals (rather than modified accelerometer signals).

[0101] The system generates one or more negative pairs of embeddings (610). In particular, for each negative pair of accelerometer signals, the system processes the first accelerometer signal using the encoder subnetwork to generate an embedding of the first accelerometer signal, and the system processes the second accelerometer signal to generate an embedding of the second accelerometer signal. The embedding of the first accelerometer signal and the embedding of the second accelerometer signal jointly define a negative pair of embeddings. In some implementations, the system generates the embeddings of the accelerometer signals using both the encoder subnetwork and an embedding neural network, as described above.

[0102] The system trains the encoder subnetwork to optimize an auxiliary contrastive objective function that depends on the positive pairs of embeddings and the negative pairs of embeddings (612). More specifically, the auxiliary contrastive objective function can, for each pair of embeddings, measure an error (e.g., a Euclidean distance) between the first embedding and the second embedding. In particular, for each positive pair of embeddings, the auxiliary contrastive objective function can encourage a higher similarity between the first embedding and the second embedding. For each negative pair of embeddings, the auxiliary contrastive objective function can encourage a lower similarity between the first embedding and the second embedding.

[0103] To train the encoder subnetwork to optimize the auxiliary contrastive objective function, the system can determine gradients of the auxiliary contrastive objective function (e.g., using backpropagation), and then adjust the parameter values of the encoder subnetwork using the gradients, e.g., in accordance with an update rule of a gradient descent optimization algorithm, e.g., RMSprop or Adam. That is, the system can backpropagate gradients of the auxiliary contrastive objective function through the encoder subnetwork. In implementations where the system generates embeddings of accelerometer signals using both the encoder subnetwork and an embedding neural network (as described above), the system can jointly train the encoder subnetwork and the embedding neural network, e.g., by backpropagating gradients through the embedding neural network and into the encoder subnetwork.

[0104] FIG. 6B provides an illustration of pre-training the encoder subnetwork to perform a contrastive embedding task, as described with reference to FIG. 6A. In particular, FIG. 6B illustrates a first transformation 616 being applied to a base accelerometer signal 614 to generate a first transformed version 618 of the base accelerometer signal 614, and a second transformation 620 being applied to the base accelerometer signal 614 to generate a second transformed version 622 of the base accelerometer signal 614. The system processes generates respective embeddings of the first and second transformed versions of the base accelerometer signal using the encoder subnetwork 202 and the embedding neural network 624. The system then jointly trains the encoder subnetwork and the embedding neural network to optimize the auxiliary contrastive objective function 628.

**[0105]** FIG. 7 is a flow diagram of an example process 700 for pre-training the encoder subnetwork of the classification neural network to perform a masked reconstruction task. For convenience, the process 700 will be described as being performed by a system of one or more computers located in one or more locations. For example, a training system, e.g., the training system 500 of FIG. 5, appropriately programmed in accordance with this specification, can perform the process 700. The training system can iteratively perform the steps of the process 700 as part of pre-training the encoder subnetwork.

**[0106]** The system obtains an accelerometer signal (702).

**[0107]** The system generates a masked accelerometer signal by masking a portion of the accelerometer signal (704). The system can mask a portion of the accelerometer signal by replacing a portion of the accelerometer signal with default data, e.g., predefined values (e.g., zero values) or random noise (e.g., sampled from a Normal distribution). The system can randomly select the portion of the accelerometer signal to be masked.

**[0108]** The system processes the masked accelerometer signal using the encoder subnetwork to generate an embedding of the masked accelerometer signal (706).

**[0109]** The system processes the embedding of the masked accelerometer signal using a decoder neural network to generate a predicted reconstruction of the accelerometer signal (708). The decoder neural network can have any appropriate neural network architecture that enables the decoder neural network to generate a predicted reconstruction of an accelerometer signal. In particular, the decoder neural network can include any appropriate types of neural network layers (e.g., convolutional layers, recurrent layers, attention layers, fully connected layers, etc.) in any number (e.g., 5 layers, 10 layers, or 50 layers) and in any appropriate configuration (e.g., as a linear sequence of layers).

**[0110]** The system jointly trains the encoder subnetwork and the decoder neural network to optimize an auxiliary reconstruction objective function that measures an error in the predicted reconstruction of the accelerometer signal (710). The error can be measured, e.g., as an $L_1$ error, or as an $L_2$ error, or using any other appropriate error metric.

**[0111]** FIG. 8 is a flow diagram of an example process 800 for pre-training the encoder subnetwork of the classification neural network to perform a noisy reconstruction task. For convenience, the process 800 will be described as being performed by a system of one or more computers located in one or more locations. For example, a training system, e.g., the training system 500 of FIG. 5, appropriately programmed in accordance with this specification, can perform the process 800. The training system can iteratively perform the steps of the process 800 as part of pre-training the encoder subnetwork.

**[0112]** The system obtains an accelerometer signal (802).

**[0113]** The system generates a noised accelerometer signal by adding noise the accelerometer signal (804). The system can sample the noise from a predefined probability distribution, e.g., a Normal distribution.

**[0114]** The system processes the noised accelerometer signal using the encoder subnetwork to generate an embedding of the noised accelerometer signal (806).

**[0115]** The system processes the embedding of the noised accelerometer signal using a decoder neural network to generate a predicted de-noised accelerometer signal, i.e., a predicted reconstruction of the original (de-noised) accelerometer signal (808). The decoder neural network can have any appropriate neural network architecture that enables the decoder neural network to generate a predicted de-noised accelerometer signal. In particular, the decoder neural network can include any appropriate types of neural network layers (e.g., convolutional layers, recurrent layers, attention layers, fully connected layers, etc.) in any number (e.g., 5 layers, 10 layers, or 50 layers) and in any appropriate configuration (e.g., as a linear sequence of layers).

**[0116]** The system jointly trains the encoder subnetwork and the decoder neural network to optimize an auxiliary de-noising objective function that measures an error in the de-noised accelerometer signal (810). More specifically, the auxiliary de-noising objective function can measure an error between: (i) the original accelerometer signal, and (ii) the predicted de-noised accelerometer signal. The error can be measured, e.g., as an $L_1$ error, or as an $L_2$ error, or using any other appropriate error metric.

**[0117]** FIG. 9 is a flow diagram of an example process 900 for pre-training the encoder subnetwork of the classification neural network to perform a supervised auxiliary task. For convenience, the process 900 will be described as being performed by a system of one or more computers located in one or more locations. For example, a training system, e.g., the training system 500 of FIG. 5, appropriately programmed in accordance with this specification, can perform the process 900. The training system can iteratively perform the steps of the process 900 as part of pre-training the encoder subnetwork.

**[0118]** The system obtains: (i) an accelerometer signal, and (ii) a target label for the accelerometer signal (902). In some implementations, the target label for the accelerometer signal defines a number of steps taken by the subject in a duration of time covered by the accelerometer signal. In some implementations, the target label for the accelerometer signal defines an action performed by the subject in a duration of time covered by the accelerometer signal.

**[0119]** The system processes the accelerometer signal using the encoder subnetwork to generate an embedding of the accelerometer signal (904).

**[0120]** The system processes the embedding of the accelerometer signal using a prediction neural network to generate a prediction output that characterizes a predicted label for the accelerometer signal (906). In some implementations, the prediction output directly defines the predicted label. In some implementations, the prediction output defines a score distribution over a set of possible labels.

**[0121]** The system jointly trains the encoder subnetwork and the prediction neural network to optimize an auxiliary supervised objective function that measures an error between: (i) the target label, and (ii) the prediction output characterizing the predicted label (908). The auxiliary supervised objective function can measure the error, e.g., as a cross-entropy error, or as a squared error.

**[0122]** FIG. 10 is a flow diagram of an example process 1000 for training the classification neural network to perform a sleep-wake classification task. For convenience, the process 1000 will be described as being performed by a system of one or more computers located in one or more locations. For example, a training system, e.g., the training system 500 of FIG. 5, appropriately programmed in accordance with this specification, can perform the process 1000. The training system can iteratively perform the steps of the process 1000 as part of training the classification neural network.

**[0123]** The system obtains: (i) a network input that includes an accelerometer signal (and, optionally, one or more other types of data, e.g., cardiovascular data, audio data, video data, etc.) and (ii) a target sleep-wake classification corresponding to the network input (1002).

**[0124]** The system processes the network input using the encoder subnetwork to generate an embedding of the network input (1004).

**[0125]** The system processes the embedding of the network input using the projection subnetwork to generate a score distribution over the set of sleep-wake classes (1006).

**[0126]** The system trains the projection subnetwork of the classification neural network to optimize a sleep-wake objective function that measures an error between: (i) the score distribution over the set of sleep-wake classes, and (ii) the target sleep-wake classification (1008). The sleep-wake objective function can measure the error, e.g., as a cross-entropy error.

[0127] In particular, the system determines gradients of the sleep-wake objective function, e.g., using backpropagation, and then adjusts the values of the set of parameters of the projection subnetwork using the gradients by an update rule of an appropriate gradient descent optimization technique, e.g., RMSprop or Adam. That is, the system backpropagates gradients of the sleep-wake objective function through the projection subnetwork. In some implementations, the system jointly trains the projection subnetwork and the encoder subnetwork by backpropagating gradients through the projection subnetwork and into the encoder subnetwork. In other implementations, the system freezes the parameter values of the encoder subnetwork after pre-training the encoder subnetwork to perform one or more auxiliary tasks, and does not adjust the parameter values of the encoder subnetwork to optimize the sleep-wake objective function.

[0128] FIG. 11 is a flow diagram of an example process 1100 for training the classification neural network to perform a medical condition classification task. For convenience, the process 1100 will be described as being performed by a system of one or more computers located in one or more locations. For example, a training system, e.g., the training system 500 of FIG. 5, appropriately programmed in accordance with this specification, can perform the process 1100. The training system can iteratively perform the steps of the process 1100 as part of training the classification neural network.

[0129] The system obtains: (i) a network input that includes an accelerometer signal (and, optionally, one or more other types of data, e.g., cardiovascular data, audio data, video data, etc.) and (ii) a target medical condition classification corresponding to the network input (1102).

[0130] The system processes the network input using the encoder subnetwork to generate an embedding of the network input (1104).

[0131] The system processes the embedding of the network input using a second projection subnetwork to generate a score defining a likelihood that the subject has the medical condition (1106).

[0132] The system trains the second projection subnetwork of the classification neural network to optimize a medical condition objective function that measures an error between: (i) the score defining the likelihood that the subject has the medical condition, and (ii) the target medical condition classification (1108). The medical condition objective function can measure the error, e.g., as a cross-entropy error.

[0133] The system determines gradients of the medical condition objective function, e.g., using backpropagation, and then adjusts the values of the set of parameters of the second projection subnetwork using the gradients by an update rule of an appropriate gradient descent optimization technique, e.g., RMSprop or Adam. That is, the system backpropagates gradients of the medical condition objective function through the second projection subnetwork. In some implementations, the system jointly trains the second projection subnetwork and the encoder subnetwork by backpropagating gradients through the second projection subnetwork and into the encoder subnetwork. In other implementations, the system freezes the parameter values of the encoder subnetwork after pre-training the encoder subnetwork to perform one or more auxiliary tasks, and does not adjust the parameter values of the encoder subnetwork to optimize the medical condition objective function.

**[0134]** FIG. 12A shows experimental results that illustrate the effects of pre-training the encoder neural network of the classification neural network. Bar chart 1202 compares the balanced accuracy of a classification neural network on the sleep-wake classification task when the encoder subnetwork is not pre-trained ("supervised") and when the encoder subnetwork is pre-trained ("self-supervised"). Bar chart 1204 compares the macro-averaged F1 score of a classification neural network on the sleep-wake classification task when the encoder subnetwork is not pre-trained ("supervised") and when the encoder subnetwork is pre-trained ("self-supervised"). Each bar chart shows experimental results when the classification neural network is trained on all the labeled training data ("100% training patients") and when the classification neural network is trained on half the labeled training data ("50% training patients"). Pre-training generally improves the performance of the classification neural network, and the increase in performance is particularly pronounced in the regime where only half the labeled training data is used for training.

**[0135]** FIG. 12B shows experimental results that illustrate the sleep-wake classification accuracy of the system described in this specification ("System #3 (Ours)) in comparison to two other sleep-wake classification systems: "System #1" and "System #2". The system described in this specification outperforms both the other systems used for comparison.

**[0136]** This specification uses the term "configured" in connection with systems and computer program components. For a system of one or more computers to be configured to perform particular operations or actions means that the system has installed on it software, firmware, hardware, or a combination of them that in operation cause the system to perform the operations or actions. For one or more computer programs to be configured to perform particular operations or actions means that the one or more programs include instructions that, when executed by data processing apparatus, cause the apparatus to perform the operations or actions.

**[0137]** Embodiments of the subject matter and the functional operations described in this specification can be implemented in digital electronic circuitry, in tangibly-embodied computer software or firmware, in computer hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Embodiments of the subject matter described in this specification can be implemented as one or more computer programs, i.e., one or more modules of computer program instructions encoded on a tangible non-transitory storage medium for execution by, or to control the operation of, data processing apparatus. The computer storage medium can be a machine-readable storage device, a machine-readable storage substrate, a random or serial access memory device, or a combination of one or more of them. Alternatively or in addition, the program instructions can be encoded on an artificially-generated propagated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus for execution by a data processing apparatus.

**[0138]** The term “data processing apparatus” refers to data processing hardware and encompasses all kinds of apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can also be, or further include, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit). The apparatus can optionally include, in addition to hardware, code that creates an execution environment for computer programs, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them.

**[0139]** A computer program, which may also be referred to or described as a program, software, a software application, an app, a module, a software module, a script, or code, can be written in any form of programming language, including compiled or interpreted languages, or declarative or procedural languages; and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A program may, but need not, correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data, e.g., one or more scripts stored in a markup language document, in a single file dedicated to the program in question, or in multiple coordinated files, e.g., files that store one or more modules, sub-programs, or portions of code. A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a data communication network.

**[0140]** In this specification the term "engine" is used broadly to refer to a software-based system, subsystem, or process that is programmed to perform one or more specific functions. Generally, an engine will be implemented as one or more software modules or components, installed on one or more computers in one or more locations. In some cases, one or more computers will be dedicated to a particular engine; in other cases, multiple engines can be installed and running on the same computer or computers.

**[0141]** The processes and logic flows described in this specification can be performed by one or more programmable computers executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by special purpose logic circuitry, e.g., an FPGA or an ASIC, or by a combination of special purpose logic circuitry and one or more programmed computers.

**[0142]** Computers suitable for the execution of a computer program can be based on general or special purpose microprocessors or both, or any other kind of central processing unit. Generally, a central processing unit will receive instructions and data from a read-only memory or a random access memory or both. The essential elements of a computer are a central processing unit for performing or executing instructions and one or more memory devices for storing instructions and data. The central processing unit and the memory can be supplemented by, or incorporated in, special purpose logic circuitry. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical disks. However, a computer need not have such devices. Moreover, a computer can be embedded in another device, e.g., a mobile telephone, a personal digital assistant (PDA), a mobile audio or video player, a game console, a Global Positioning System (GPS) receiver, or a portable storage device, e.g., a universal serial bus (USB) flash drive, to name just a few.

**[0143]** Computer-readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks.

**[0144]** To provide for interaction with a user, embodiments of the subject matter described in this specification can be implemented on a computer having a display device, e.g., a CRT (cathode ray tube) or LCD (liquid crystal display) monitor, for displaying information to the user and a keyboard and a pointing device, e.g., a mouse or a trackball, by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input. In addition, a computer can interact with a user by sending documents to and receiving documents from a device that is used by the user; for example, by sending web pages to a web browser on a user's device in response to requests received from the web browser. Also, a computer can interact with a user by sending text messages or other forms of message to a personal device, e.g., a smartphone that is running a messaging application, and receiving responsive messages from the user in return.

**[0145]** Data processing apparatus for implementing machine learning models can also include, for example, special-purpose hardware accelerator units for processing common and compute-intensive parts of machine learning training or production, i.e., inference, workloads.

**[0146]** Machine learning models can be implemented and deployed using a machine learning framework, e.g., a TensorFlow framework, or a Jax framework.

**[0147]** Embodiments of the subject matter described in this specification can be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface, a web browser, or an app through which a user can interact with an implementation of the subject matter described in this specification, or any combination of one or more such back-end, middleware, or front-end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network (LAN) and a wide area network (WAN), e.g., the Internet.

**[0148]** The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other. In some embodiments, a server transmits data, e.g., an HTML page, to a user device, e.g., for purposes of displaying data to and receiving user input from a user interacting with the device, which acts as a client. Data generated at the user device, e.g., a result of the user interaction, can be received at the server from the device.

[0149] While this specification contains many specific implementation details, these should not be construed as limitations on the scope of any invention or on the scope of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular inventions. Certain features that are described in this specification in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially be claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[0150] Similarly, while operations are depicted in the drawings and recited in the claims in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system modules and components in the embodiments described above should not be understood as requiring such separation in all embodiments, and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products.

[0151] Particular embodiments of the subject matter have been described. Other embodiments are within the scope of the following claims. For example, the actions recited in the claims can be performed in a different order and still achieve desirable results. As one example, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In some cases, multitasking and parallel processing may be advantageous.

What is claimed is:

CLAIMS

1. A method performed by one or more computers, the method comprising:

performing a sleep-wake classification task using a sleep-wake classification neural network, comprising:

receiving accelerometer data generated by a wearable device of a subject;

processing the accelerometer data using a sleep-wake classification neural network, comprising:

processing the accelerometer data using an encoder subnetwork of the sleep-wake classification neural network to generate an embedding of the accelerometer data in a latent space; and

processing the embedding of the accelerometer data using a projection subnetwork of the sleep-wake classification neural network to generate a network output that defines a score distribution over a set of sleep-wake classes,

wherein the set of sleep-wake classes includes: (i) at least one class corresponding to a state of wakefulness, and (ii) at least one class corresponding to a state of sleep;

classifying a sleep-wake state of the subject based on the respective score for each class in the set of sleep-wake classes;

wherein the sleep-wake classification neural network has been trained by operations comprising:

pre-training the encoder subnetwork of the sleep-wake classification neural network to perform an auxiliary task, wherein the auxiliary task is different than the sleep-wake classification task; and

after pre-training the encoder subnetwork, training the sleep-wake classification neural network to perform the sleep-wake classification task.

2. The method of claim 1, wherein pre-training the encoder subnetwork of the sleep-wake classification neural network to perform the auxiliary task comprises:

obtaining a base accelerometer signal;

generating a positive pair of embeddings in the latent space, comprising:

processing a first transformed version of the base accelerometer signal using the encoder subnetwork to generate a corresponding embedding in the latent space;

processing a second transformed version of the base accelerometer signal using the encoder subnetwork to generate a corresponding embedding in the latent space;

training the encoder subnetwork to optimize an auxiliary objective function that depends on: (i) the embedding of the first transformed version of the base accelerometer signal, and (ii) the embedding of the second transformed version of the base accelerometer signal.

3. The method of claim 2, further comprising generating the first transformed version of the base accelerometer signal, comprising:

randomly sampling a first transformation from a space of transformations; and

applying the first transformation to the base accelerometer signal to generate the first transformed version of the base accelerometer signal.

4. The method of any one of claims 2-3, further comprising generating the second transformed version of the base accelerometer signal, comprising:

randomly sampling a second transformation from a space of transformations; and

applying the second transformation to the base accelerometer signal to generate the second transformed version of the base accelerometer signal.

5. The method of any one of claims 2-4, wherein the auxiliary objective function measures an error between: (i) embedding of the first transformed version of the base accelerometer signal, and (ii) the embedding of the second transformed version of the base accelerometer signal.

6. The method of any one of claims 2-5, wherein pre-training the encoder subnetwork of the sleep-wake classification neural network to perform the auxiliary task comprises:

obtaining an accelerometer signal;

generating a masked accelerometer signal by masking a portion of the accelerometer signal;

processing the masked accelerometer signal using the encoder subnetwork to generate an embedding of the masked accelerometer signal;

processing the embedding of the masked accelerometer signal using a decoder neural network to generate a predicted reconstruction of the accelerometer signal; and

jointly training the encoder subnetwork and the decoder neural network to optimize an auxiliary objective function that measures an error in the predicted reconstruction of the accelerometer signal.

7. The method of any one of claims 2-6, wherein pre-training the encoder subnetwork of the sleep-wake classification neural network to perform the auxiliary task comprises:

obtaining an accelerometer signal;

generating a noised accelerometer signal by adding noise the accelerometer signal;

processing the noised accelerometer signal using the encoder subnetwork to generate an embedding of the noised accelerometer signal;

processing the embedding of the noised accelerometer signal using a decoder neural network to generate a de-noised accelerometer signal;

jointly training the encoder subnetwork and the decoder neural network to optimize an auxiliary objective function that measures an error in the de-noised accelerometer signal.

8. The method of any one of claims 2-7, wherein pre-training the encoder subnetwork of the sleep-wake classification neural network to perform the auxiliary task comprises:

obtaining: (i) an accelerometer signal, and (ii) a target label for the accelerometer signal;

processing the accelerometer signal using the encoder subnetwork to generate an embedding of the accelerometer signal;

processing the embedding of the accelerometer signal using a prediction neural network to generate a predicted label for the accelerometer signal; and

jointly training the encoder subnetwork and the decoder neural network to optimize an auxiliary objective function that measures an error between: (i) the target label, and (ii) the predicted label.

9. The method of claim 8, wherein the target label for the accelerometer signal defines a number of steps taken by the subject in a duration of time covered by the accelerometer signal.

10. The method of claim 8, wherein the target label for the accelerometer signal defines an action performed by the subject in a duration of time covered by the accelerometer signal.

11. The method of any preceding claim, wherein training the sleep-wake classification neural network to perform the sleep-wake classification task comprises:

obtaining: (i) a training accelerometer signal, and (ii) a target sleep-wake classification for the training accelerometer signal;

processing the training accelerometer signal using the encoder subnetwork to generate an embedding of the training accelerometer signal;

processing the embedding of the training accelerometer signal using the projection subnetwork to generate a score distribution over the set of sleep-wake classes; and

training the projection subnetwork of the sleep-wake classification neural network to optimize a sleep-wake objective function that measures an error between: (i) the score distribution over the set of sleep-wake classes, and (ii) the target sleep-wake classification.

12. The method of claim 11, wherein parameter values of the encoder subnetwork are frozen during the training of the sleep-wake classification neural network to perform the sleep-wake classification task.

13. The method of any preceding claim, wherein training the sleep-wake classification neural network to perform the sleep-wake classification task comprises:

applying dropout to one or more layers of the sleep-wake classification neural network during the training to perform the sleep-wake classification task.

14. The method of any preceding claim, wherein the sleep-wake classification neural network includes one or more residual blocks, wherein each residual block is configured to:

receive a block input;

process the block input, by one or more neural network layers of the residual block, to generate an intermediate output; and

generate a block output by summing: (i) the block input, and (ii) the intermediate output.

15. The method of any preceding claim, wherein the set of sleep-wake classes includes multiple classes corresponding to respective states of sleep.

16. The method of claim 15, wherein the set of sleep-wake classes includes a class corresponding to rapid eye movement (REM) sleep.

17. The method any one of claims 15-16, wherein the set of sleep-wake classes includes one or more classes corresponding to non-REM sleep.

18. The method of claim 17, wherein the set of sleep-wake classes includes respective classes corresponding to one or more of: stage 1 non-REM sleep, stage 2 non-REM sleep, or stage 3 non-REM sleep.

19. The method of any preceding claim, wherein performing the sleep-wake classification task further comprises receiving cardiovascular data for the subject; and

wherein processing the accelerometer data using the sleep-wake classification neural network comprises:

jointly processing the accelerometer data and the cardiovascular data using the sleep-wake classification neural network.

20. The method of any preceding claim, wherein performing the sleep-wake classification task further comprises receiving ambient light data for the surroundings of the subject; and

wherein processing the accelerometer data using the sleep-wake classification neural network comprises:

jointly processing the accelerometer data and the ambient light data using the sleep-wake classification neural network.

21. The method of any preceding claim, wherein performing the sleep-wake classification task further comprises receiving audio data characterizing sound in the surroundings of the subject; and

wherein processing the accelerometer data using the sleep-wake classification neural network comprises:

jointly processing the accelerometer data and the audio data using the sleep-wake classification neural network.

22. The method of any preceding claim, wherein performing the sleep-wake classification task further comprises receiving time data characterizing a current time of day when the

accelerometer data is captured; and

wherein processing the accelerometer data using the sleep-wake classification neural network comprises:

jointly processing the accelerometer data and the time data using the sleep-wake classification neural network.

23. The method of any preceding claim, wherein classifying the sleep-wake state of the subject based on the respective score for each class in the set of sleep-wake classes comprises:

classifying the sleep-wake state of the subject into a class associated with a highest score from among the set of sleep-wake classes.

24. The method of any preceding claim, further comprising:

repeatedly performing the sleep-wake classification over a time window to generate a set of sleep-wake classifications; and

processing the set of sleep-wake classifications to predict a duration of time, during the time window, that the subject was in a state of sleep.

25. The method of any preceding claim, further comprising, in response to classifying the sleep-wake state of the subject as being a sleep state:

triggering an alarm to cause the subject to transition into a state of wakefulness.

26. A system comprising:

one or more computers; and

one or more storage devices communicatively coupled to the one or more computers, wherein the one or more storage devices store instructions that, when executed by the one or more computers, cause the one or more computers to perform operations of the respective method of any one of claims 1-25.

27. One or more non-transitory computer storage media storing instructions that when executed by one or more computers cause the one or more computers to perform operations of the respective method of any one of claims 1-25.

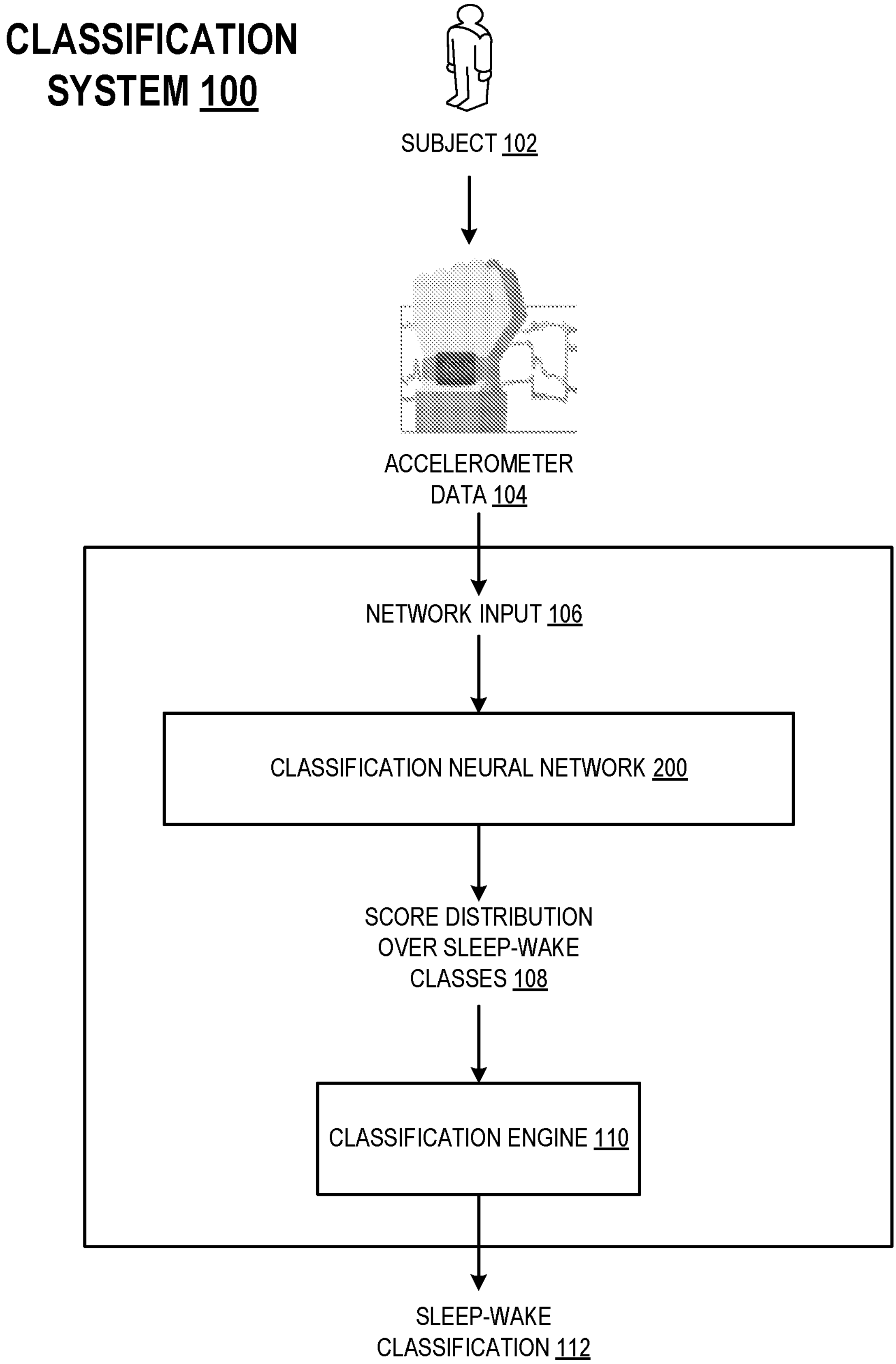
CLASSIFICATION SYSTEM 100
SUBJECT 102
ACCELEROMETER DATA 104
NETWORK INPUT 106
CLASSIFICATION NEURAL NETWORK 200
SCORE DISTRIBUTION OVER SLEEP-WAKE CLASSES 108
CLASSIFICATION ENGINE 110
SLEEP-WAKE CLASSIFICATION 112

FIG. 1

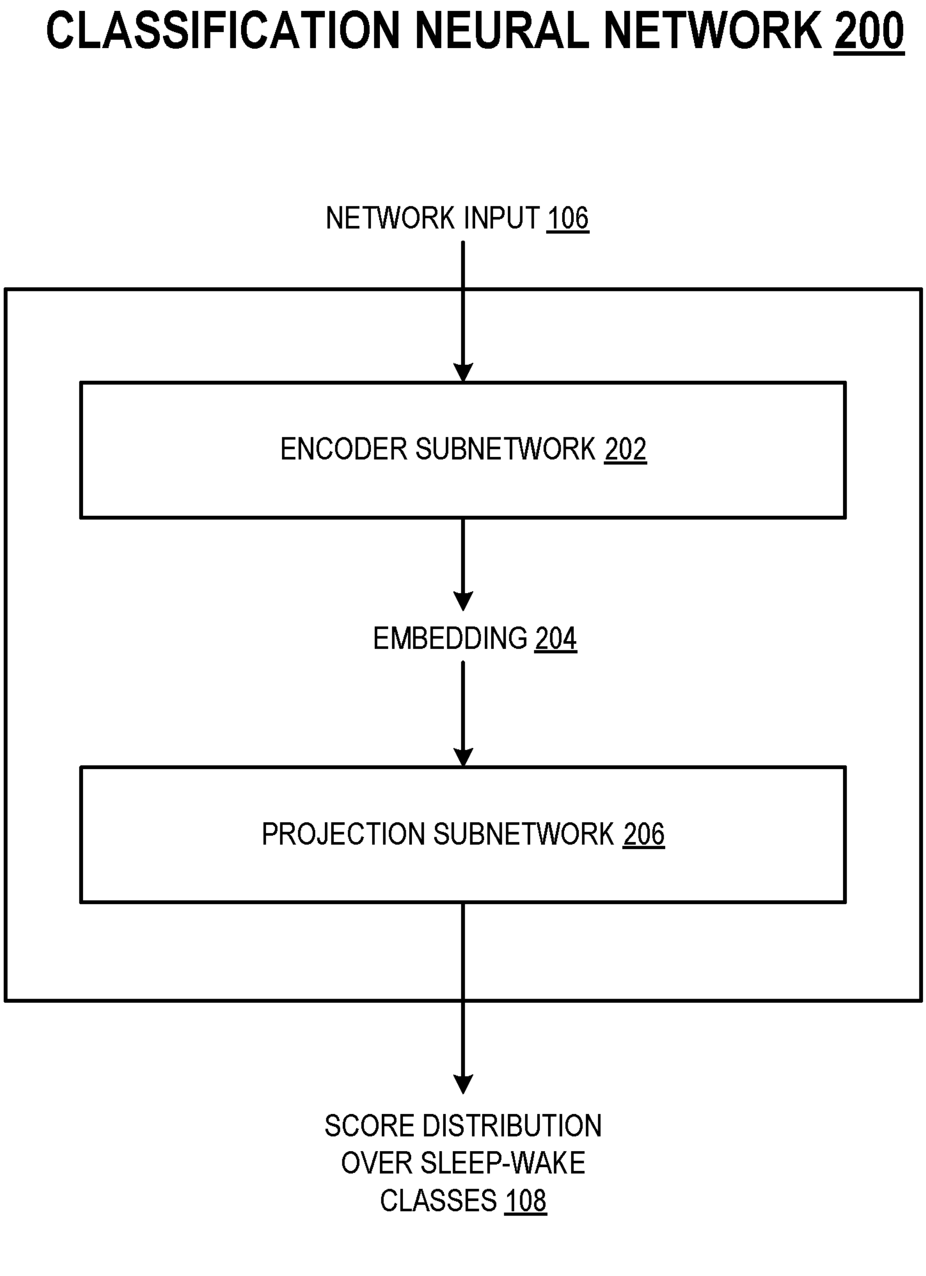
CLASSIFICATION NEURAL NETWORK 200
NETWORK INPUT 106
ENCODER SUBNETWORK 202
EMBEDDING 204
PROJECTION SUBNETWORK 206
SCORE DISTRIBUTION OVER SLEEP-WAKE CLASSES 108

FIG. 2

# CLASSIFICATION NEURAL NETWORK IMPLEMENTED USING RESIDUAL BLOCKS

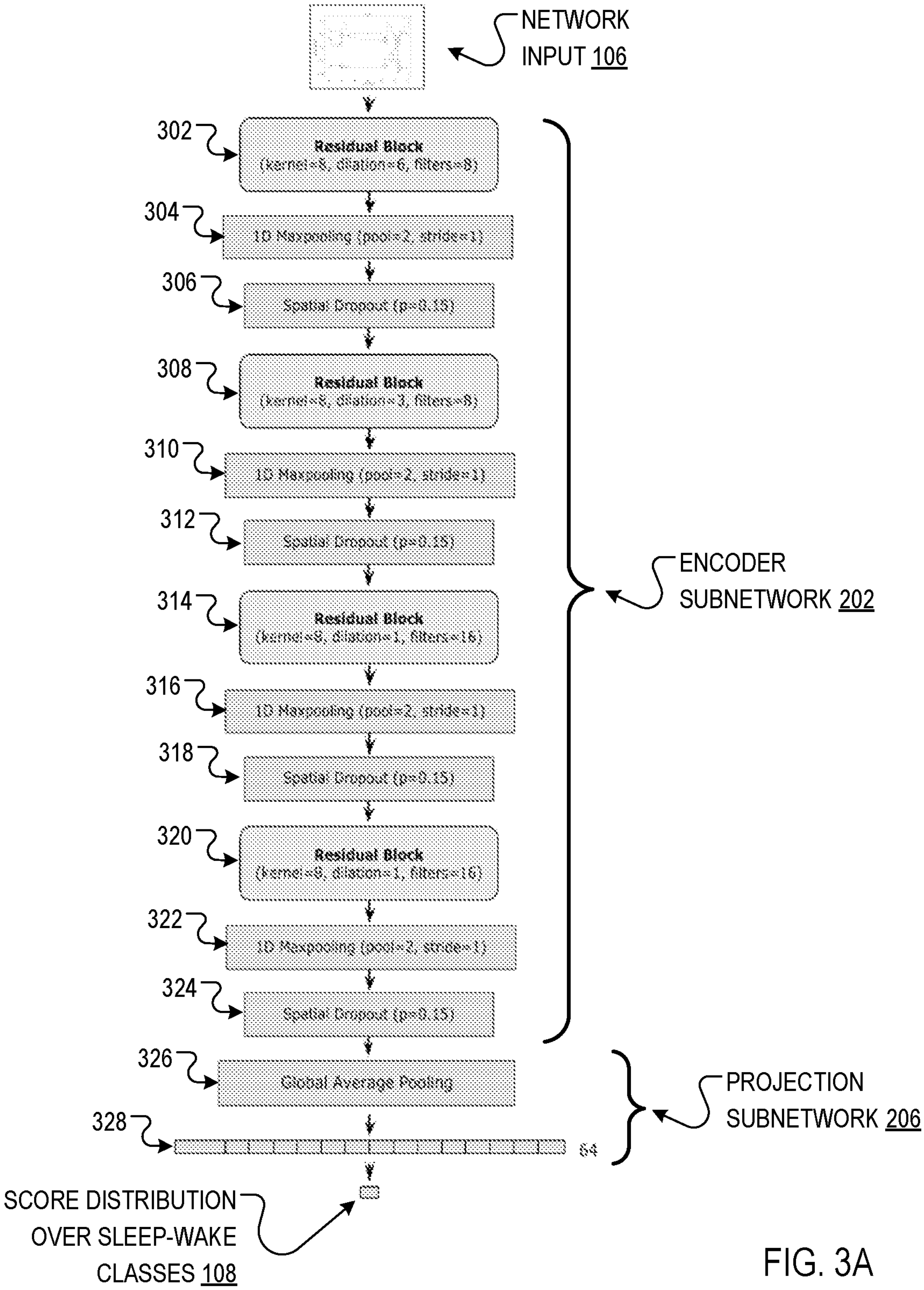


FIG. 3A

# RESIDUAL BLOCK IN CLASSIFICATION NEURAL NETWORK

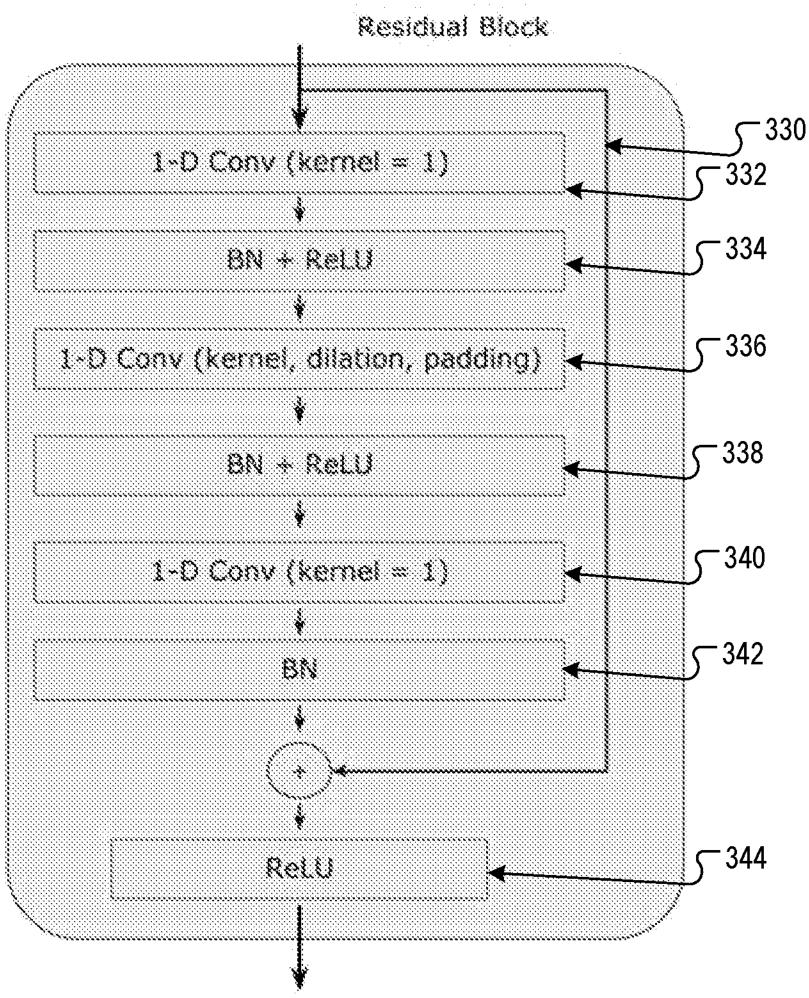


FIG. 3B

## CLASSIFYING THE SLEEP-WAKE STATE OF A SUBJECT

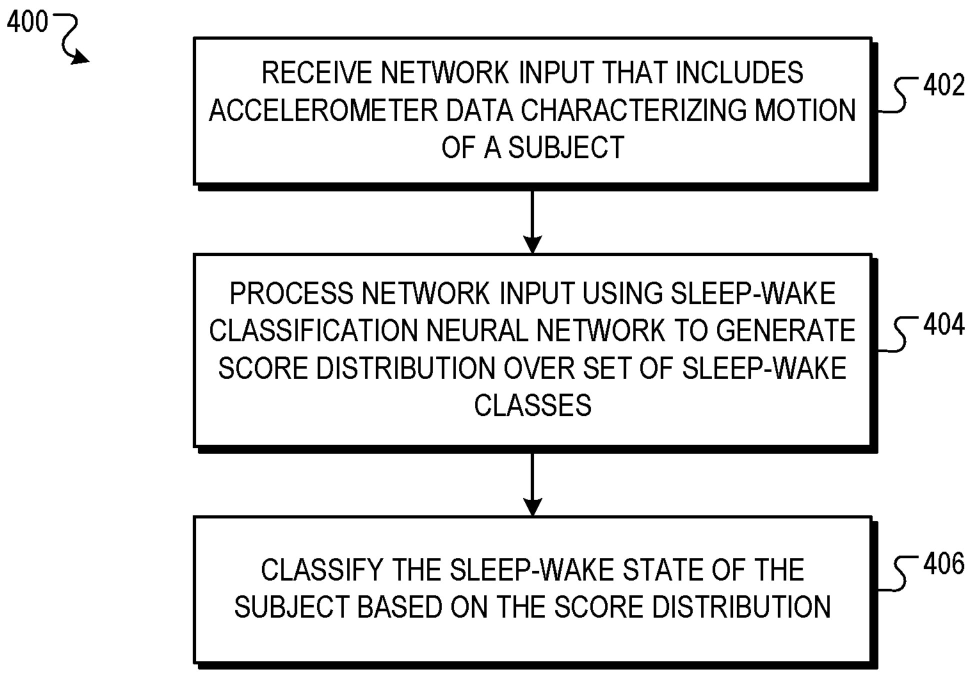


FIG. 4

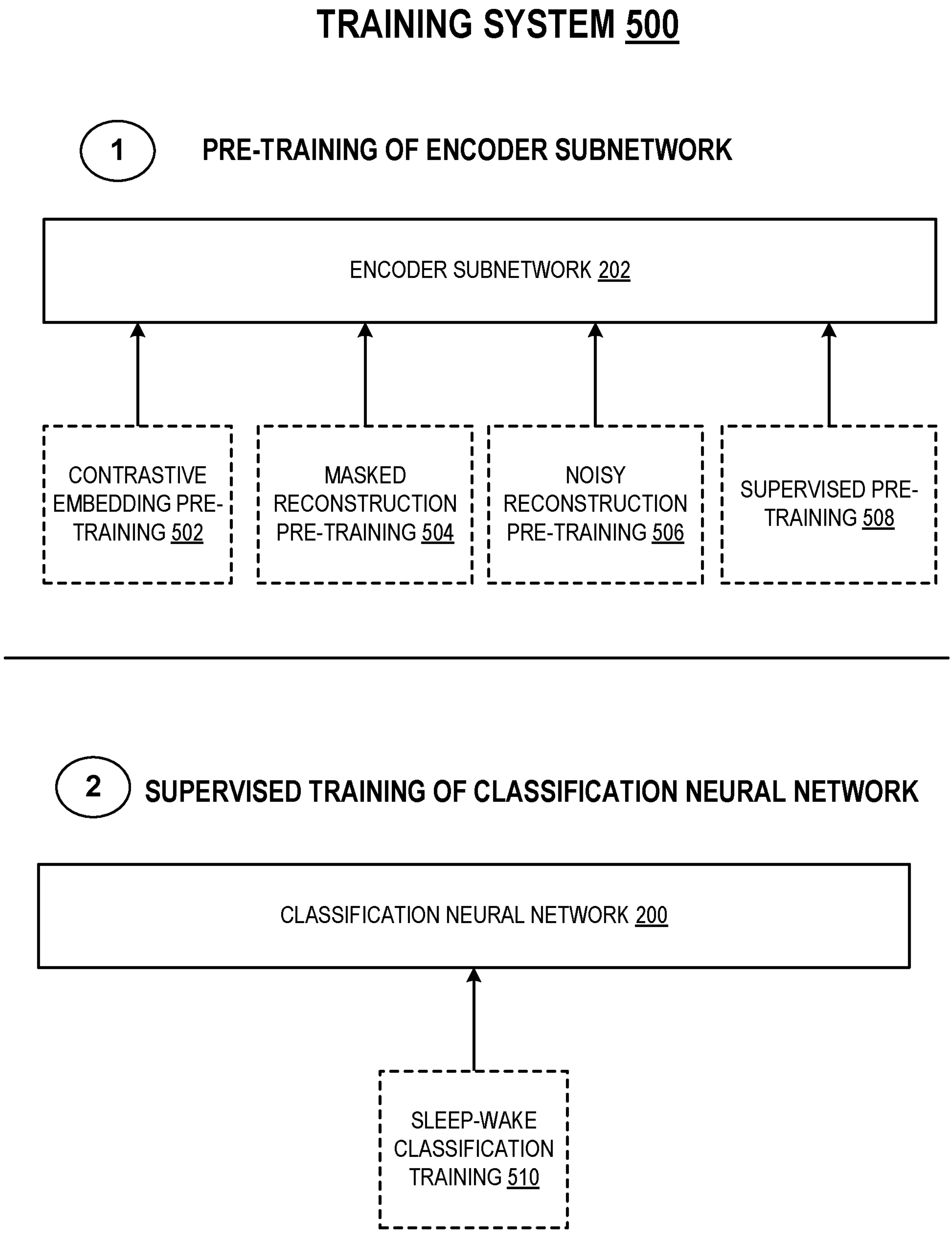
TRAINING SYSTEM 500
1
PRE-TRAINING OF ENCODER SUBNETWORK
ENCODER SUBNETWORK 202
CONTRASTIVE EMBEDDING PRE-TRAINING 502
MASKED RECONSTRUCTION PRE-TRAINING 504
NOISY RECONSTRUCTION PRE-TRAINING 506
SUPERVISED PRE-TRAINING 508
2
SUPERVISED TRAINING OF CLASSIFICATION NEURAL NETWORK
CLASSIFICATION NEURAL NETWORK 200
SLEEP-WAKE CLASSIFICATION TRAINING 510

FIG. 5

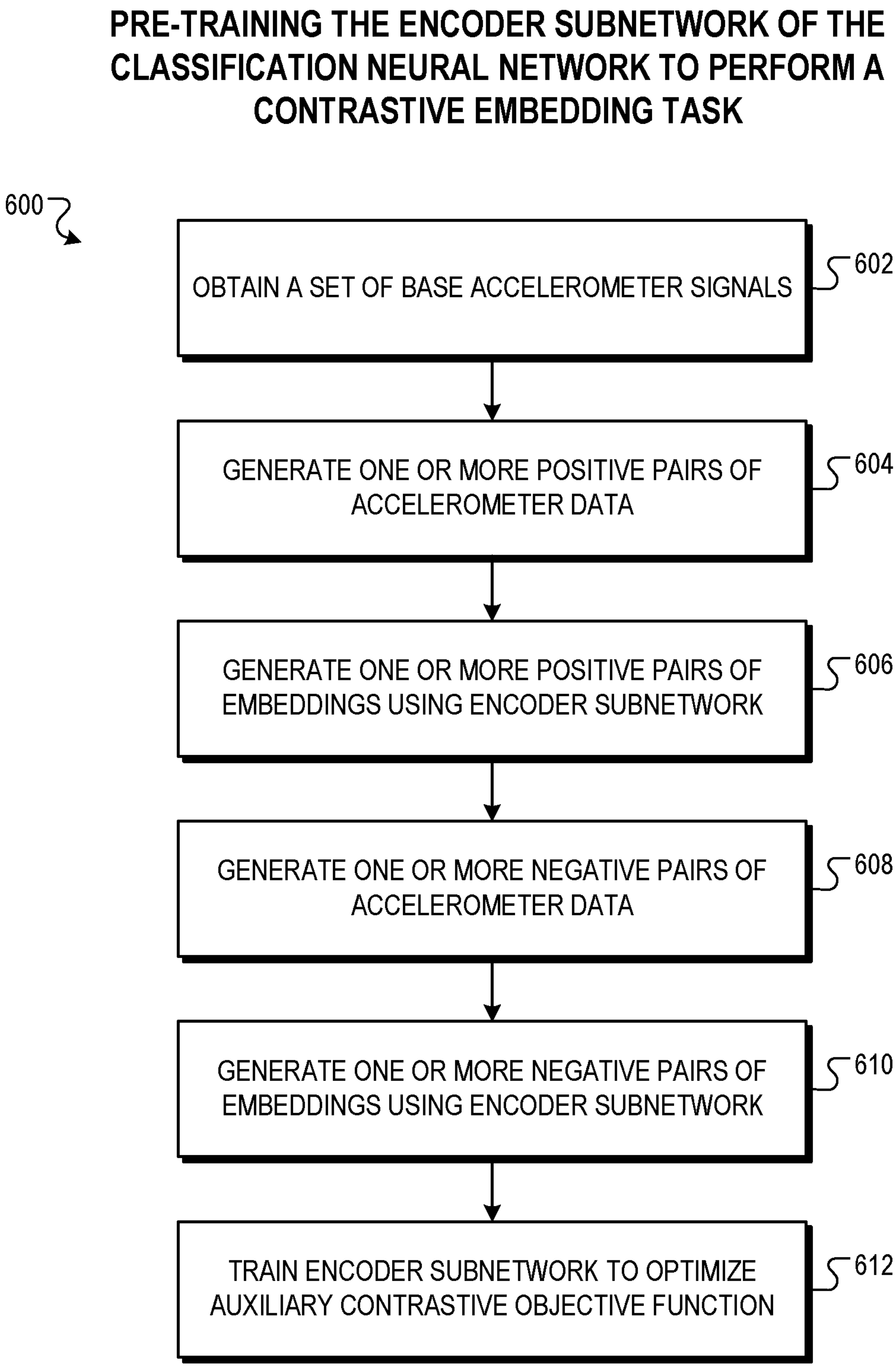
PRE-TRAINING THE ENCODER SUBNETWORK OF THE CLASSIFICATION NEURAL NETWORK TO PERFORM A CONTRASTIVE EMBEDDING TASK
600
OBTAIN A SET OF BASE ACCELEROMETER SIGNALS
602
GENERATE ONE OR MORE POSITIVE PAIRS OF ACCELEROMETER DATA
604
GENERATE ONE OR MORE POSITIVE PAIRS OF EMBEDDINGS USING ENCODER SUBNETWORK
606
GENERATE ONE OR MORE NEGATIVE PAIRS OF ACCELEROMETER DATA
608
GENERATE ONE OR MORE NEGATIVE PAIRS OF EMBEDDINGS USING ENCODER SUBNETWORK
610
TRAIN ENCODER SUBNETWORK TO OPTIMIZE AUXILIARY CONTRASTIVE OBJECTIVE FUNCTION
612

FIG. 6A

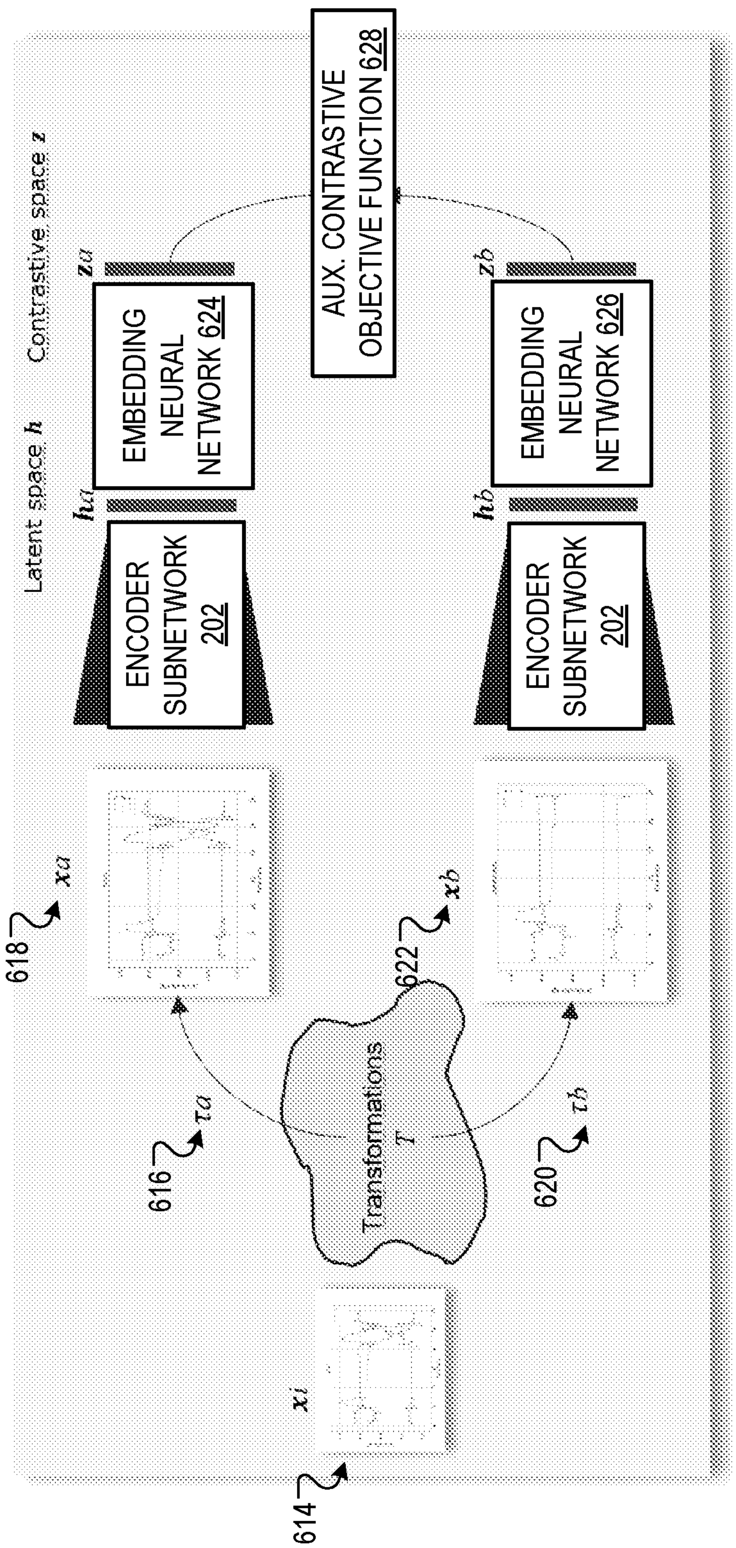
Latent space h
Contrastive space z
AUX. CONTRASTIVE OBJECTIVE FUNCTION 628
ha
za
ENCODER SUBNETWORK 202
EMBEDDING NEURAL NETWORK 624
hb
zb
ENCODER SUBNETWORK 202
EMBEDDING NEURAL NETWORK 626
618
xa
622
xb
616
τa
620
τb
Transformations
T
614
xi

FIG. 6B

PRE-TRAINING THE ENCODER SUBNETWORK OF THE CLASSIFICATION NEURAL NETWORK TO PERFORM A MASKED RECONSTRUCTION TASK

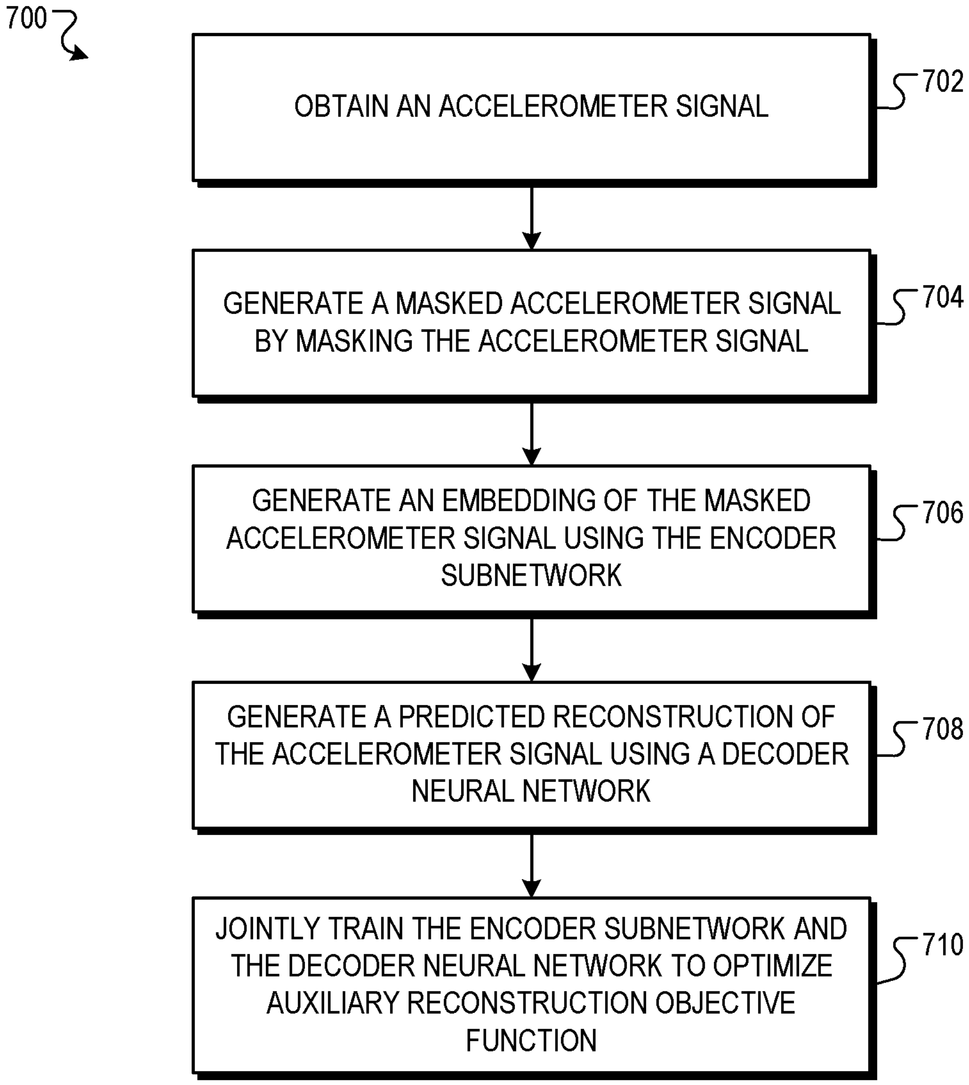


FIG. 7

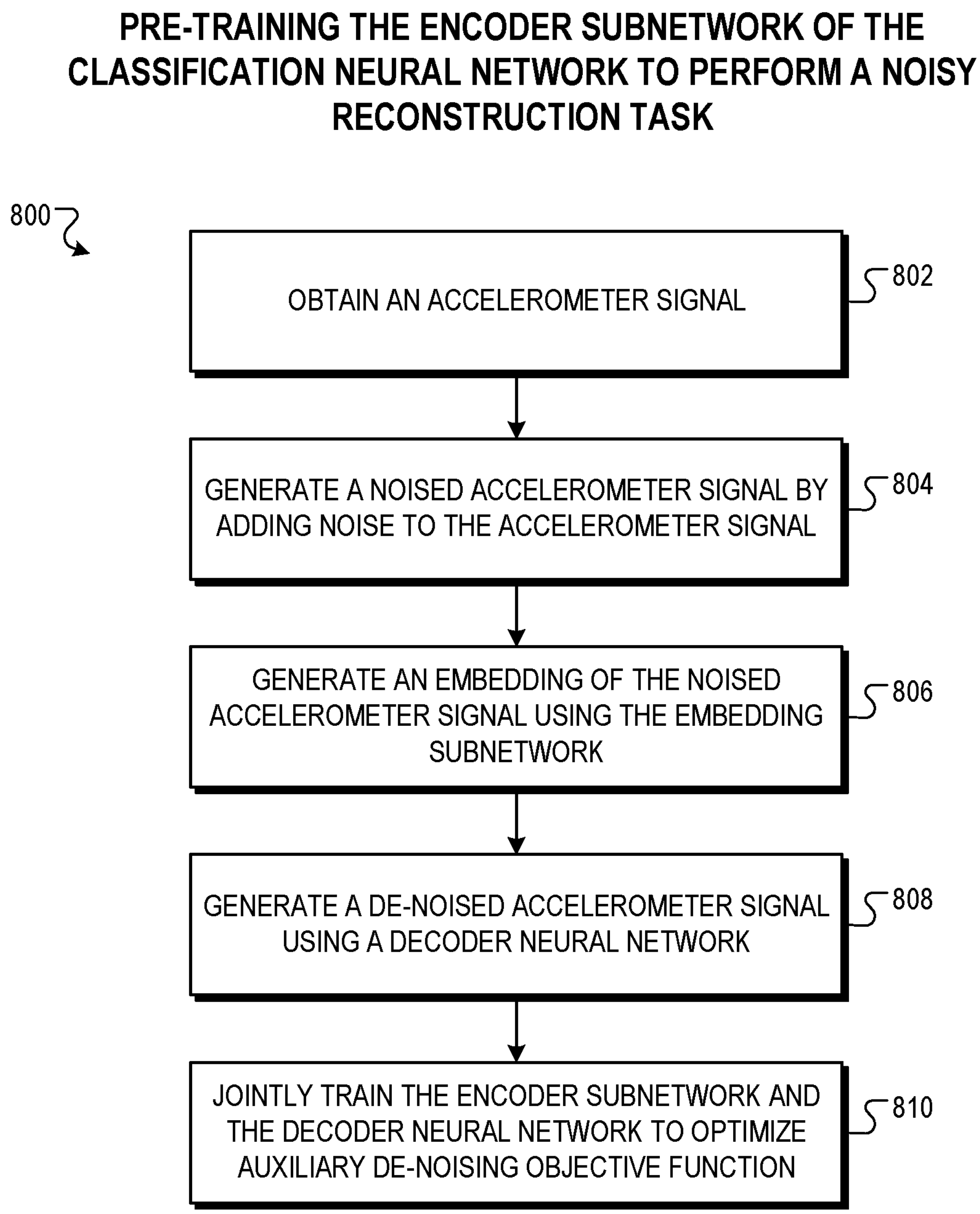
PRE-TRAINING THE ENCODER SUBNETWORK OF THE CLASSIFICATION NEURAL NETWORK TO PERFORM A NOISY RECONSTRUCTION TASK
800
OBTAIN AN ACCELEROMETER SIGNAL
802
GENERATE A NOISED ACCELEROMETER SIGNAL BY ADDING NOISE TO THE ACCELEROMETER SIGNAL
804
GENERATE AN EMBEDDING OF THE NOISED ACCELEROMETER SIGNAL USING THE EMBEDDING SUBNETWORK
806
GENERATE A DE-NOISED ACCELEROMETER SIGNAL USING A DECODER NEURAL NETWORK
808
JOINTLY TRAIN THE ENCODER SUBNETWORK AND THE DECODER NEURAL NETWORK TO OPTIMIZE AUXILIARY DE-NOISING OBJECTIVE FUNCTION
810

FIG. 8

**PRE-TRAINING THE ENCODER SUBNETWORK OF THE CLASSIFICATION NEURAL NETWORK TO PERFORM A SUPERVISED AUXILIARY TASK**

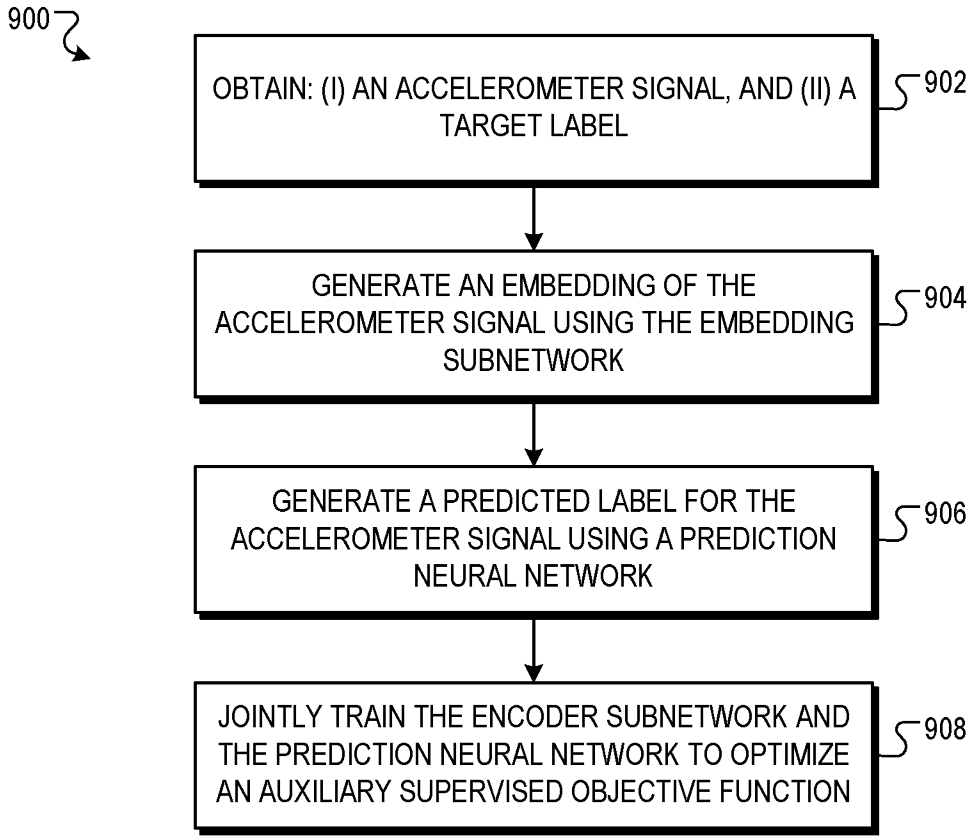


FIG. 9

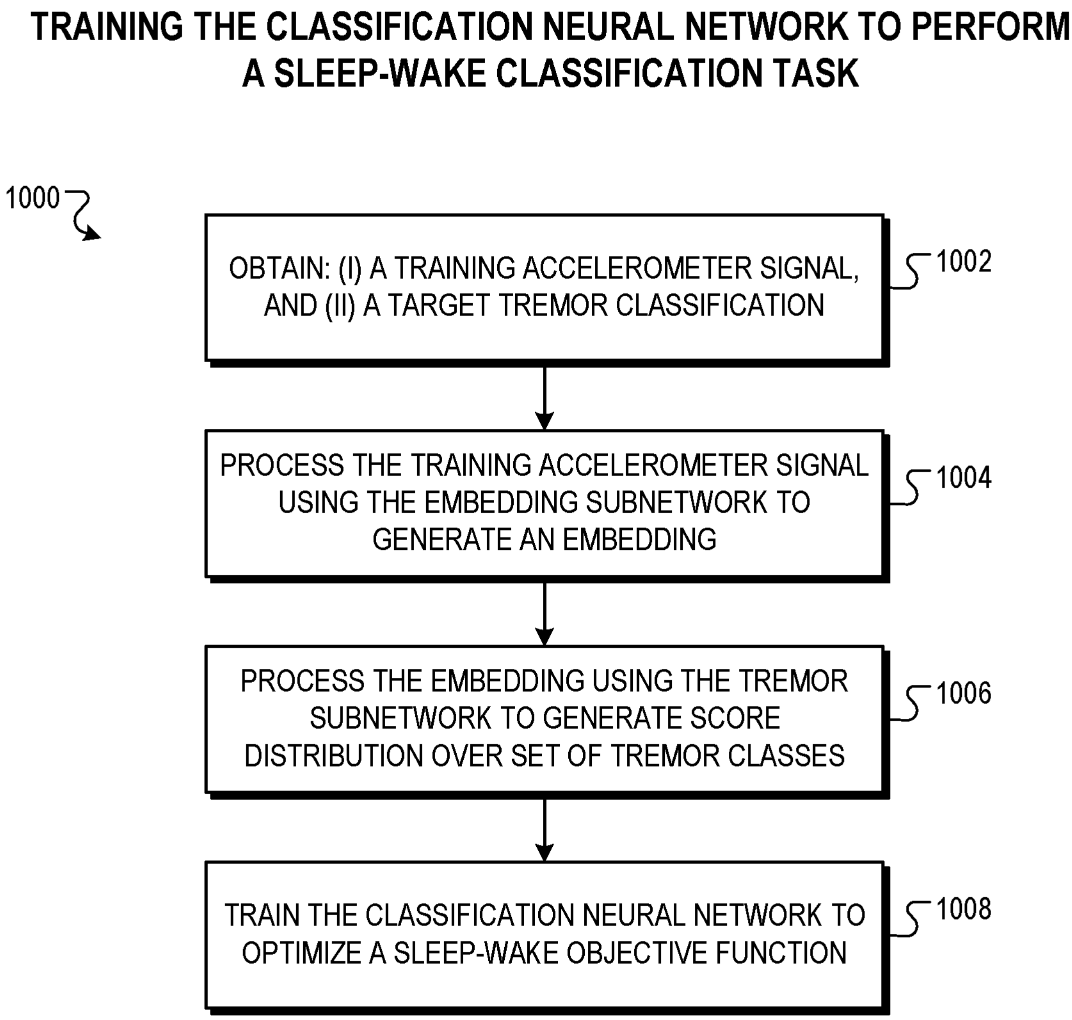
TRAINING THE CLASSIFICATION NEURAL NETWORK TO PERFORM A SLEEP-WAKE CLASSIFICATION TASK
1000
OBTAIN: (I) A TRAINING ACCELEROMETER SIGNAL, AND (II) A TARGET TREMOR CLASSIFICATION
1002
PROCESS THE TRAINING ACCELEROMETER SIGNAL USING THE EMBEDDING SUBNETWORK TO GENERATE AN EMBEDDING
1004
PROCESS THE EMBEDDING USING THE TREMOR SUBNETWORK TO GENERATE SCORE DISTRIBUTION OVER SET OF TREMOR CLASSES
1006
TRAIN THE CLASSIFICATION NEURAL NETWORK TO OPTIMIZE A SLEEP-WAKE OBJECTIVE FUNCTION
1008

FIG. 10

TRAINING THE CLASSIFICATION NEURAL NETWORK TO PERFORM A MEDICAL CONDITION CLASSIFICATION TASK

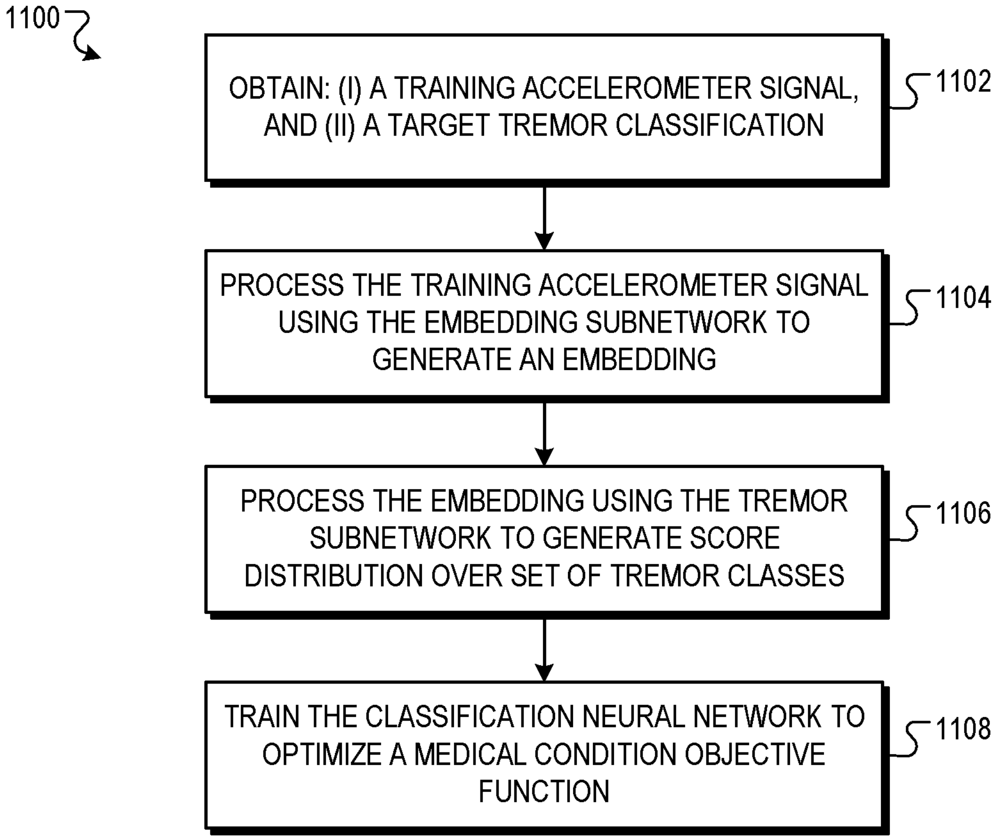


FIG. 11

EXPERIMENTAL RESULTS: EFFECTS OF PRE-TRAINING

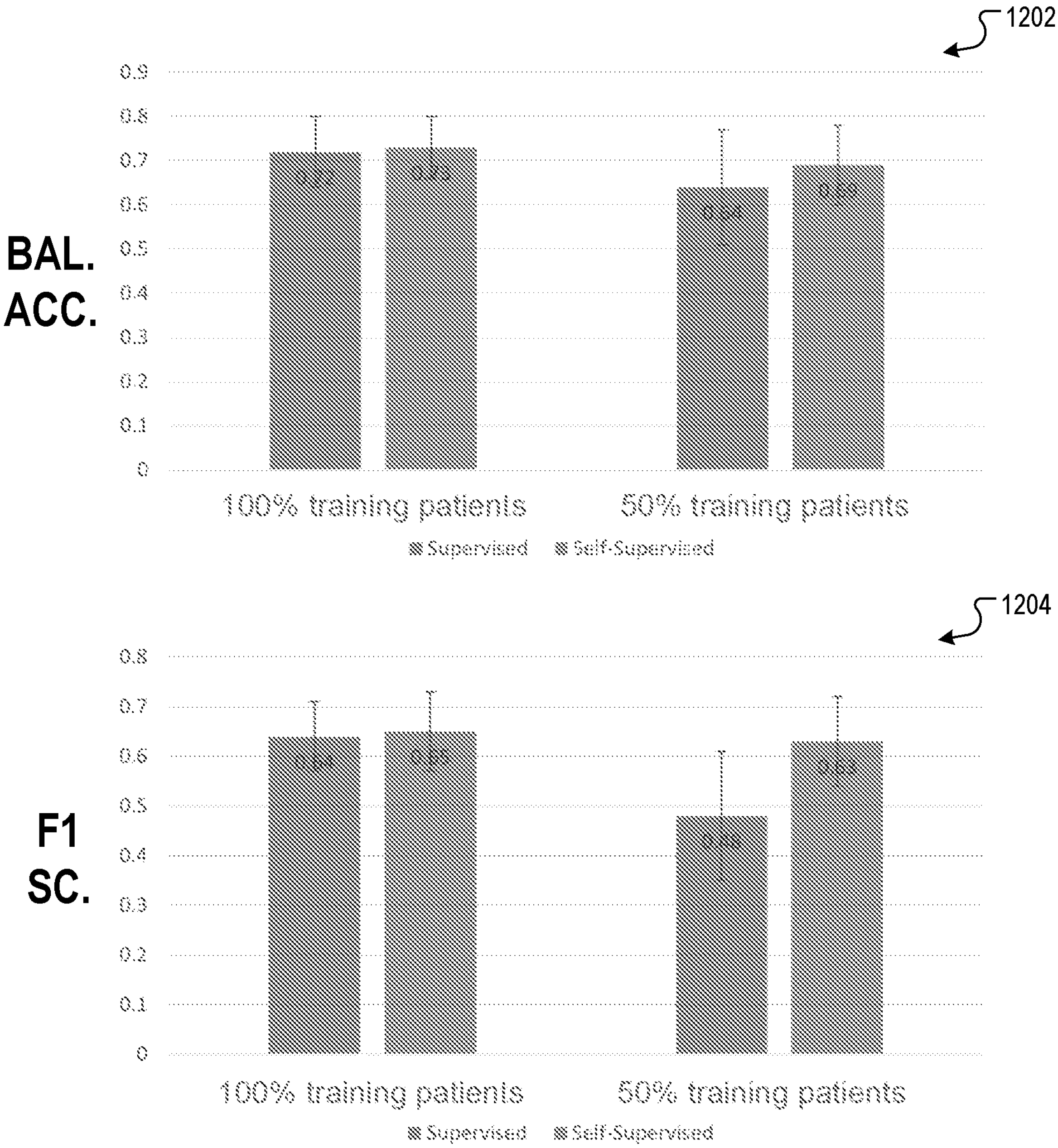


FIG. 12A

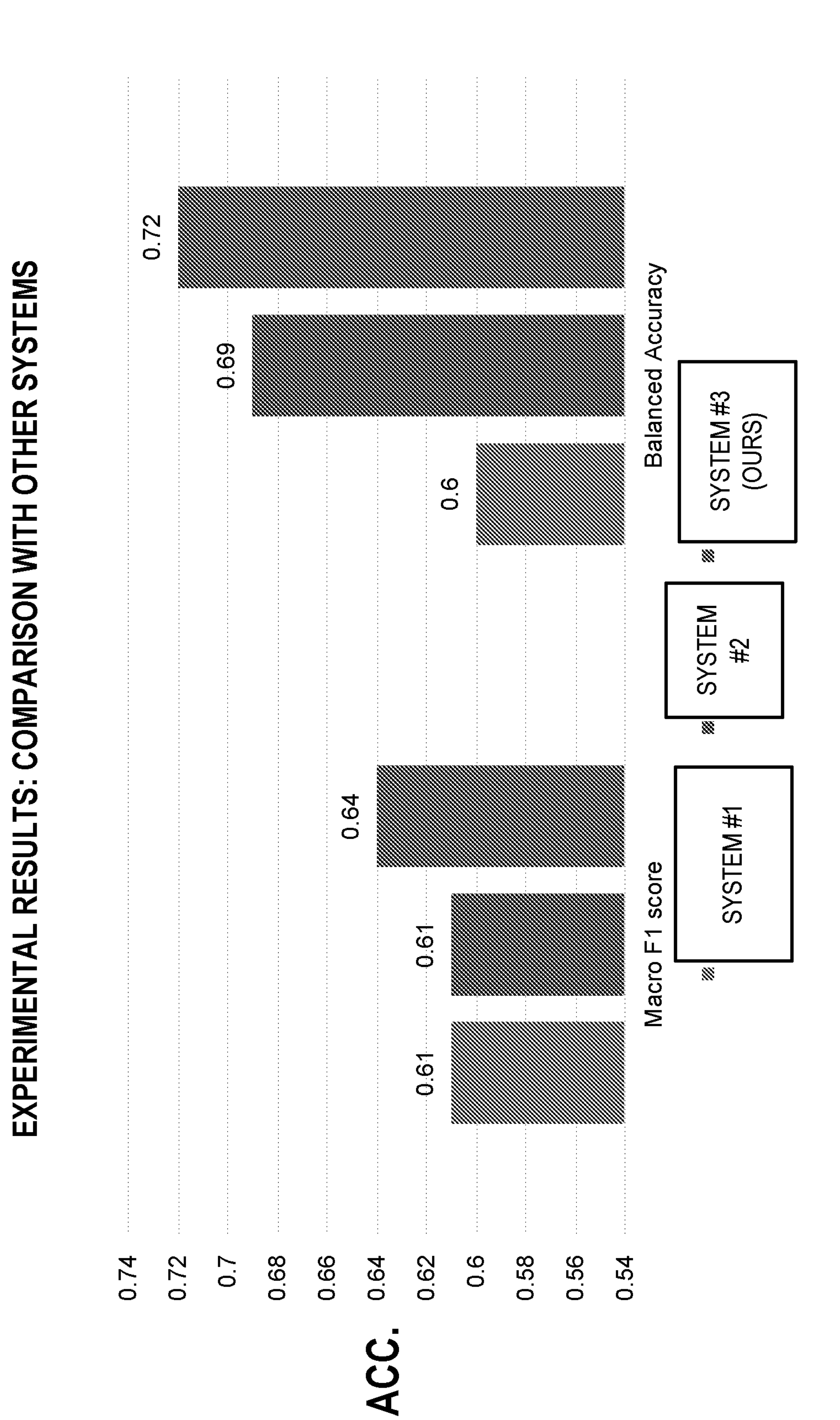
EXPERIMENTAL RESULTS: COMPARISON WITH OTHER SYSTEMS
ACC.
0.74
0.72
0.7
0.68
0.66
0.64
0.62
0.6
0.58
0.56
0.54
0.61
0.61
0.64
0.6
0.69
0.72
Macro F1 score
Balanced Accuracy
SYSTEM #1
SYSTEM #2
SYSTEM #3 (OURS)

FIG. 12B

# INTERNATIONAL SEARCH REPORT

International application No
PCT/US2024/031274

**A. CLASSIFICATION OF SUBJECT MATTER**
INV. G16H50/20 A61B5/00 G06N3/045
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)
G16H G06N A61B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| Y | CHEN ZHENGHUA ET AL: "An Attention Based CNN-LSTM Approach for Sleep-Wake Detection With Heterogeneous Sensors", IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, IEEE, PISCATAWAY, NJ, USA, [Online] vol. 25, no. 9, 30 June 2020 (2020-06-30), pages 3270-3277, XP011876424, ISSN: 2168-2194, DOI: 10.1109/JBHI.2020.3006145 [retrieved on 2021-09-02] the whole document in particular sections II-IV; figure 1; table II<br>-----<br>-/-- | 1-27 |

[x] Further documents are listed in the continuation of Box C.

[x] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 30 August 2024 | 16/09/2024 |

| Name and mailing address of the ISA/ | Authorized officer |
|---|---|
| European Patent Office, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040,<br>Fax: (+31-70) 340-3016 | Rákossy, Z |

1

# INTERNATIONAL SEARCH REPORT

International application No
PCT/US2024/031274

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| Y | AAQIB SAEED ET AL: "Sense and Learn: Self-Supervision for Omnipresent Sensors", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, [Online] 6 September 2021 (2021-09-06), XP091044131, [retrieved on 2024-08-29] the whole document in particular sections 3, 4; figure 2 | 1-27 |
| Y | US 2023/107505 A1 (BUNEL RUDY [GB] ET AL) 6 April 2023 (2023-04-06) the whole document in particular paragraph [0005] - paragraph [0021]; figure 1 | 1-27 |
| X | SHOHREH DELDARI ET AL: "COCOA: Cross Modality Contrastive Learning for Sensor Data", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, [Online] 3 August 2022 (2022-08-03), XP091287061, DOI: 10.1145/3550316 [retrieved on 2024-08-29] | 1-27 |
| Y | the whole document in particular sections 3,4; figures 1,2,8 | 1-27 |
| X,P | CHANG KAI CHIEH ET AL: "Classification of Infant Sleep/Wake States: Cross-Attention among Large Scale Pretrained Transformer Networks using Audio, ECG, and IMU Data", 2023 ASIA PACIFIC SIGNAL AND INFORMATION PROCESSING ASSOCIATION ANNUAL SUMMIT AND CONFERENCE (APSIPA ASC), IEEE, [Online] 31 October 2023 (2023-10-31), pages 2370-2377, XP034471916, DOI: 10.1109/APSIPAASC58517.2023.10317201 [retrieved on 2023-11-20] | 1,10,11, 15-22, 25-27 |
| Y,P | the whole document in particular sections II-IV; figures 1,2 | 2-9,11, 12,23,24 |

1

# INTERNATIONAL SEARCH REPORT

**Information on patent family members**

International application No

PCT/US2024/031274

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| US 2023107505 A1 | 06-04-2023 | CN 115190999 A | 14-10-2022 |
| | | EP 4094198 A1 | 30-11-2022 |
| | | US 2023107505 A1 | 06-04-2023 |
| | | WO 2021247962 A1 | 09-12-2021 |