

IPR2024-01234

Patent Owner's Preliminary Response

Filed on behalf of FriendliAI Inc.

By:

JENNIFER R. BUSH, Reg. No 50,784

JAE WON SONG, Reg. No. 59,070

JAE YEON BAEK, Reg. No. 78,258

MICHAEL J. SACKSTEDER (*pro hac vice*)

FENWICK & WEST LLP

801 California Street

Mountain View, CA 94041

Telephone: 650.988.8500

Facsimile: 650.938.5200

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

HUGGING FACE, INC.

Petitioner,

v.

FRIENDLIAI INC.

Patent Owner.

Case No. IPR2024-01234

Patent 11,442,775

PATENT OWNER'S PRELIMINARY RESPONSE

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. BACKGROUND	3
A. The '775 Patent	3
B. Prosecution History of the '775 Patent	7
III. Claim Construction.....	9
IV. STATE OF THE ART	10
A. Recurrent Neural Networks (RNNs).....	10
B. Transformers	14
C. Batching in RNNs vs. Transformer Models.....	19
D. Dynamic Batching for Transformer Models	25
V. overview of the prior art	26
A. Katharopoulos	26
B. Gao	31
VI. PETITIONER FAILED TO SHOW IT IS MORE LIKELY THAN NOT THAT THE CLAIMS ARE INVALID UNDER 35 U.S.C. §103	33
A. Petitioner Has Not Met Its Statutory Burden	33
1. Citation Errors Foreclose Required “Particularity” Showing	34
2. The Petition Improperly Relies On Non-analogous Art	35
3. The Petition Did Not Establish Public Availability of the Katharopoulos Reference.....	38

TABLE OF CONTENTS
(Continued)

	Page
B. Defects in Obviousness Analysis	40
1. The Proposed Combination Suffers from Fatal Flaws for Failure to Show Certain Claim Elements.....	42
a. The proposed combination does not meet the claim limitations that explicitly recite “running,” “access[ing],” and “applying” a transformer model (1[pre], 1[a], 1[c], 1[f]; 10[pre], 10[a], 10[c], 10[f]).....	42
(1) 1[pre]	44
(2) 1[a].....	45
(3) 1[c].....	45
(4) 1[f]	46
b. The combined references do not disclose or suggest “the second batch of requests scheduled responsive to determining that the execution engine has memory available to execute the second batch of requests” (1[e])	47
c. The Petition Fails to Establish a <i>Prima Facie</i> Case of Obviousness.....	48
2. Petitioner Fails to Prove a Motivation to Combine	49
a. A POSITA would not select and combine these references as proposed.....	50
(1) A POSITA faced with the problem of processing a batch of requests with variable lengths in transformers would not select the Gao and Katharopoulos references.....	51

TABLE OF CONTENTS
(Continued)

	Page
(2) A POSITA would not have combined references in a way that eliminates the transformer model entirely	51
(3) A POSITA would not be motivated to modify a state-of-the-art transformer model to approximate to a less sophisticated RNN model.....	53
(4) A POSITA would not be motivated to modify the references as proposed	54
b. Petitioner provides no valid rationale for the combination	56
(1) Petitioner's rationale neglects to consider the disadvantages of the combination	56
(2) Petitioner relies on fictional, unsupported rationales	57
(3) Petitioner's expert's parroted testimony cannot save its faulty analysis	59
(4) Hindsight is the only possible rationale to combine Gao and Katharopoulos	60
C. Dr. Hammouri's Testimony Should Be Afforded Little Weight	61
VII. CONCLUSION.....	62

TABLE OF AUTHORITIES

	Page(s)
CASES	
<i>A.C. Dispensing Equip. Inc. v. Prince Castle LLC</i> , IPR2014-00511, Paper 16 (P.T.A.B. Sep. 10, 2014).....	35
<i>Acceleration Bay, LLC v. Activision Blizzard, Inc.</i> , 908 F.3d 765 (Fed. Cir. 2018)	38, 39
<i>ActiveVideo Networks, Inc. v. Verizon Commc'ns, Inc.</i> , 694 F.3d 1312 (Fed. Cir. 2012)	61, 62
<i>Amazon Web Servs., Inc. et al. v. Saint Regis Mohawk Tribe</i> , IPR2019-00103, Paper 22 (P.T.A.B. May 10, 2019)	35
<i>Apple Inc. v. Samsung Elecs. Co.</i> , 839 F.3d 1034 (Fed. Cir. 2016) (en banc)	41, 48
<i>CFMT, Inc. v. YieldUp Int'l Corp.</i> , 349 F.3d 1333	42
<i>Corning Incorp. v. DDSM IP Assets B.V.</i> , IPR2013-00043, Paper 95 (P.T.A.B. May 1, 2014)	34
<i>DeSilva v. DiLeonardi</i> , 181 F.3d 865 (7th Cir. 1999)	34, 58
<i>Dynaenergetics US, Inc. v. Geodynamics, Inc.</i> , PGR2018-00065, Paper 8 (P.T.A.B. Nov. 15, 2018)	39
<i>Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.</i> , 800 F.3d 1375 (Fed. Cir. 2015)	41
<i>Epistar Corp. v. Int'l Trade Comm'n</i> , 566 F.3d 1321 (Fed. Cir. 2009)	10
<i>First Solar, Inc. v. Rovshan Sade</i> , IPR2023-00827, Paper 13 (P.T.A.B. Nov. 16, 2023).....	2, 4, 40
<i>Gamber-Johnson LLC v. Nat'l Products Inc.</i> , IPR2021-01159, Paper 7 (P.T.A.B. Jan. 3, 2022)	49, 57

TABLE OF AUTHORITIES
(Continued)

	Page(s)
<i>Garmin Int'l, Inc. v. Patent of Cuozzo Speed Techs. LLC</i> , IPR2012-00001, Paper 15 (P.T.A.B. Jan. 9, 2013)	42
<i>Graham v. John Deere Co. of Kansas City</i> , 383 U.S. 1 (1966).....	41, 48
<i>Harmonic Inc. v. Avid Tech., Inc.</i> , 815 F.3d 1356 (Fed. Cir. 2016)	41
<i>Hulu, LLC, v. Sound View Innovations, LLC</i> , IPR2018-01039, Paper 29 (P.T.A.B. Dec. 20, 2019)	38
<i>In re Bigio</i> , 381 F.3d 1320 (Fed. Cir. 2004)	35, 36
<i>In re Clay</i> , 966 F.2d 656 (Fed. Cir. 1992)	37
<i>In re Klein</i> , 647 F.3d 1343 (Fed. Cir. 2011)	36
<i>In re Magnum Oil Tools Int'l, Ltd.</i> , 829 F.3d 1364 (Fed. Cir. 2016)	41, 55, 60
<i>In re Ratti</i> , 270 F.2d 810 (C.C.P.A. 1959)	52
<i>In re Rijckaert</i> , 9 F.3d 1531 (Fed. Cir. 1993)	42
<i>In re Wood</i> , 599 F.2d 1032 (C.C.P.A. 1979)	36
<i>In-Depth Geophysical, Inc. v. Conocophillips Co.</i> , IPR2019-00849, Paper 14 (P.T.A.B. Sept. 6, 2019).....	39
<i>Innovention Toys, LLC, v. MGA Ent., Inc.</i> , 637 F.3d 1314 (Fed. Cir. 2011)	35

TABLE OF AUTHORITIES
(Continued)

	Page(s)
<i>KSR Int'l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	41, 49, 50
<i>Merck & Cie v. Gnosis S.p.A.</i> , 808 F.3d 829 (Fed. Cir. 2015)	56
<i>Meta Platforms, et al. v. VL Collective IP LLC</i> , IPR2023-00924, Paper 11 (P.T.A.B. Dec. 6. 2023)	39
<i>Metalcraft of Mayville, Inc. v. The Toro Co.</i> , 848 F.3d 1358 (Fed. Cir. 2017)	61
<i>Novartis AG v. Torrent Pharms. Ltd.</i> , 853 F.3d 1316 (Fed. Cir. 2017)	49
<i>Purdue Pharma L.P. v. Depomed, Inc.</i> , IPR2014-00377, Paper 17 (P.T.A.B. Aug. 6, 2014).....	35
<i>Retractable Techs., Inc. v. Becton, Dickinson & Co.</i> , 653 F.3d 1296 (Fed. Cir. 2011)	10
<i>Rohm & Haas Co. v. Brotech Corp.</i> , 127 F.3d 1089 (Fed. Cir. 1997)	60, 62
<i>Ruiz v. A.B. Chance Co.</i> , 357 F.3d 1270 (Fed. Cir. 2004)	61
<i>ServiceNow Inc., v. Hewlett-Packard Co.</i> , IPR2015-00716, Paper 13 (P.T.A.B. Aug. 26, 2015).....	40
<i>SFC Co. v. LG Chem Ltd.</i> , IPR2020-00178, 2020 WL 2790521 (P.T.A.B. May 29, 2020)	60
<i>SRI Int'l, Inc. v. Internet Sec. Sys.</i> , 511 F.3d 1186 (Fed. Cir. 2008)	40
<i>Unigene Labs., Inc. v. Apotex, Inc.</i> , 655 F.3d 1352 (Fed. Cir. 2011)	50, 51

TABLE OF AUTHORITIES
(Continued)

	Page(s)
<i>Winner Int'l Royalty Corp. v. Wang</i> , 202 F.3d 1340 (Fed. Cir. 2000)	56
<i>Xerox Corp. v. Bytemark, Inc.</i> , IPR2022-00624, Paper 12 (P.T.A.B. Feb. 10, 2023).....	60, 61, 62
STATUTES AND RULES	
35 U.S.C. §103	<i>passim</i>
35 U.S.C. §112(b)	9
35 U.S.C. §312(a)(3).....	33, 34, 41
35 U.S.C. §316(e)	33
OTHER AUTHORITIES	
37 C.F.R. §42.107(a).....	1
MPEP §2143.01	51

EXHIBIT LIST (37 CFR §42.63(e))

Exhibit	Description
2001	Expert Declaration of Michael Pazzani
2002	A Critical Review of Recurrent Neural Networks for Sequence Learning, Zachary C. Lipton, et al., June 5, 2015
2003	E-BATCH: Energy-Efficient and High-Throughput RNN Batching, Franyell Silfa, et al., Universitat Politècnica de Catalunya, September 22, 2020
2004	Sequences of Unequal Length, Chapter 7, Introduction to Deep Learning, Sandro Skansi, 2018
2005	Learning to (Learn at Test Time): RNNs with Expressive Hidden States, Yu Sun, et al., July 5, 2024
2006	An Introduction to Transformers, Richard E. Turner, February 8, 2024
2007	FastSeq: Make Sequence Generation Faster, Yu Yan, et al., July 12, 2021
2008	From Turing to Transformers: A Comprehensive Review and Tutorial on the Evolution and Applications of Generative Transformer Models, Emma Yann Zhang, et al., <i>Sci</i> , 2023
2009	Serial order: A parallel distributed processing approach, Michael I. Jordan. Technical Report 8604, Institute for Cognitive Science, University of California, San Diego, 1986
2010	Declaration of Michael J. Sacksteder in Support of Patent Owner's Motion for Pro Hac Vice Admission Pursuant to 37 C.F.R. §42.10(c)
2011	Finding structure in time, Jeffrey L. Elman. Cognitive science, 14(2):179 1990

Exhibit	Description
2012	Revised Joint Claim Construction Chart (Dkt. 109), <i>FriendlyAI Inc. v. Hugging Face, Inc.</i> , United States District Court, District of Delaware, Case No. 1:23-CV-00816-MN
2013	Joint Claim Construction Brief (Public Version) (Dkt. 125), <i>FriendlyAI Inc. v. Hugging Face, Inc.</i> , United States District Court, District of Delaware, Case No. 1:23-CV-00816-MN
2014	Stipulation and Order Amending Case Schedule (Dkt. 86), <i>FriendlyAI Inc. v. Hugging Face, Inc.</i> , United States District Court, District of Delaware, Case No. 1:23-CV-00816-MN
2015	What does it mean by Transformers are RNNs, Sandesh Ghimir Blog post: Aug. 8, 2024 (“Ghimire”)
2016	cosFormer: Rethinking Softmax in Attention, Qin et al., from ICLR 2022.
2017	Long Short-Term Memory, Sepp Hochreiter; Jürgen Schmidhuber. Neural Computation, 9 (8): 1735-1780 (1997). Available at https://www.bioinf.jku.at/publications/older/2604.pdf
2018	Anonymous, Recurrent Linear Transformers, ICLR 2024, available at https://openreview.net/pdf?id=dALYqPm9gW
2019	RNNs are not Transformers (Yet): The Key Bottleneck on In-context Retrieval, Kaiyue Wen, et al. May 2024, available at https://openreview.net/pdf?id=dALYqPm9gW

Pursuant to 37 C.F.R. §42.107(a), Patent Owner FriendlyAI Inc. (“FriendlyAI” or “Patent Owner”) submits the following Preliminary Response to the Petition for *Inter Partes* Review of U.S. Patent No. 11,442,775 (“the ’775 Patent”) submitted by Petitioner Hugging Face, Inc. (“Hugging Face” or “Petitioner”). For the following reasons, the Board should deny institution for failure to show it is more likely than not that any of the challenged claims are unpatentable.

I. INTRODUCTION

The Board should deny institution because the Petition’s obviousness analysis is backwards and erroneous. Petitioner incorrectly argues that a person of ordinary skill in the art (POSITA)¹ would modify a machine learning transformer model, in which the claimed invention executes batches of requests is performed, into an older, inferior RNN as suggested by the Katharopoulos reference, and then combine that with the RNN-specific cellular batching as disclosed in the Gao reference, to reach the claimed invention. But a POSITA would not select and combine references directed to rewriting linear transformers as RNNs (Katharopoulos) and performing

¹ Although Patent Owner disagrees with Petitioner’s POSITA proposal as requiring slightly more education and experience than Petitioner proposes (Ex. 2001, ¶¶28-29), that difference does not impact the analysis herein. Ex. 2001, ¶30; *see also id.*, ¶¶1, 13-14, 16, 22-23, 24.

cellular batching in RNNs (Gao), because these references are not reasonably pertinent to the problem to be solved in the '775 Patent of how to process a batch of requests with variable lengths in transformer models without using more resources.

The Board should not institute due to the Petition's numerous flaws. ***First***, Petitioner has failed its statutory burden of identifying "with particularity" the evidence that supports the ground asserted for at least three reasons, each of which alone is sufficient to deny institution. The Petition is rife with citation errors to the wrong reference or wrong page number and Petitioner has not even established a reasonable likelihood that the Katharopoulos reference was publicly accessible before the critical date. The references are non-analogous art under both tests, as neither is from the same field of endeavor or reasonably pertinent to the particular problem the inventor was trying to solve.

Second, the references fail to establish a *prima facie* case of obviousness since they do not disclose or suggest several claimed limitations. The claim is directed to, and several limitations explicitly recite, being performed using a transformer model, which neither reference discloses. The references also do not disclose the recited memory size check, since batching in RNNs uses a fixed hidden state length across iterations. In addition, Gao's RNN-specific cellular batching does not apply to transformers. Each of these failures alone warrant denial of institution. The combined references pertain to RNNs, an older model than the Katharopoulos

reference itself admits uses a “fundamentally different approach” as compared to the recited transformer model, which was developed to address many of the shortcomings of RNNs.

Third, Petitioner's motivation to combine analysis has several defects, each of which alone is fatal to institution. The Petition does not even articulate the differences between the prior art and the claims as required, nor does it make any attempt to show that a POSITA would have selected and combined the relied-upon prior art elements. Even if a POSITA would combine these references, the combination would eliminate use of a transformer model entirely, defeating the principle of operation, and the resultant RNN model would require retraining, which the Petition did not acknowledge. The only imaginable rationale left for the proposed combination is improper hindsight based on the teachings of the patent itself. Institution should be denied.

II. BACKGROUND

A. The '775 Patent

In the artificial intelligence (AI) computing field, the operation of machine-learning neural networks is computationally intensive. Specifically, the '775 patent applies iteration-level batching in AI transformer models. The patent explains, “at one or more iterations, the inference system can modify the batch being executed on

the execution engine by adding new incoming requests to the batch or removing completed requests from the batch.” *Id.*, 2:62-64; *see also* Ex. 2001, ¶¶31-32.

The claimed invention overcame then-existing problems in the field, including latency and low throughput, by batching and scheduling requests at an iteration-level. This action is known as “dynamic batching” or “continuous batching.” “[D]ynamic batching allows the serving system 435 to dynamically adjust batches that are processed on the execution engines such that the hardware of the execution engine can be fully utilized.” Ex. 1001, 24:28-31. Some advantages of the invention are that “the response can be provided to the user faster,” it avoids the execution engine “being under-utilized,” and it achieves “higher processor utilization on the hardware accelerators.” *Id.*, 3:2-6, 5:30-32; *see* Ex. 2001, ¶33.

The ’775 patent enables transformer batches to have variable input length, variable target length, and variable internal state length. Ex. 1001, 1:55-61. This is not required in RNNs. In previous inference serving systems for machine-learning transformer models, the systems “require[d] the length of inputs or the lengths of internal states to the transformer model be the same” across the batch. *Id.*, 21:63-22:1. With the ’775 patent,

dynamic batching allows the serving system 435 to dynamically adjust batches that are processed on the execution engines such that the hardware of the execution engine can be fully utilized. This process was not available for

certain methods of batching (e.g., method of FIGS. 2A-2B) for transformer models, because internal states, such as the key cache tensor and value cache tensor, maintain variable length per iteration in a transformer model, while other recurrent machine-learning models (e.g., RNNs) make use of state data that have the same lengths across a batch of requests.

Id., 24:28-38; *see also* Ex. 2001, ¶¶34-36.

The claims of the '775 patent recite a method of performing dynamic batching for transformer models including “scheduling . . . a batch of requests including the one or more requests for execution on an execution engine,” “generating . . . a first set of output tokens by applying the transformer model to a first set of inputs for the batch of requests,” “scheduling . . . a second batch of requests additionally including the new request for execution on the execution engine,” and “generating . . . a second set of output tokens by applying the transformer model to the second set of inputs for the second batch.” Ex. 2001, ¶37.

Thus, the claimed method allows batches of requests to be modified between iterations, without starting and completing a batch of requests together. The length of the user prompt tokens for a batch of requests and the length of the KV Cache for the batch of requests need not be the same, neither is the use of dummy tokens required. This allows, for example, requests to be added or removed between iterations. Ex. 2001, ¶73.

For example, claim 1 of the '775 patent recites claim language directed to iteration-level scheduling (bolded and italicized) below:

1[pre] 1. A method of *dynamically executing batches of requests* on one or more execution engines running a machine learning transformer model, comprising:

1[a] receiving, by a serving system, one or more requests for execution, the serving system including a scheduler and . . . a machine-learning transformer model including at least a set of decoders;

1[b] *scheduling, by the scheduler, a batch of requests including the one or more requests for execution on an execution engine;*

1[c] *generating, by the execution engine, a first set of output tokens* by applying the transformer model to a first set of inputs for the batch of requests, wherein applying the transformer model comprises applying at least one batch operation to one or more input tensors associated with the batch of requests;

1[d] receiving, by a request processor, a new request from a client device, the new request including a sequence of input tokens;

1[e] *scheduling, by the scheduler, a second batch of requests additionally including the new request for execution on the*

execution engine, the second batch of requests scheduled responsive to determining that the execution engine has memory available to execute the second batch of requests, wherein in a second set of inputs for the second batch of requests, a length of the sequence of input tokens for the new request is different from a length of an input for at least one request other than the new request; and

1[f] *generating, by the execution engine, a second set of output tokens* by applying the transformer model to the second set of inputs for the second batch.²

Ex. 2001, ¶37.

B. Prosecution History of the '775 Patent

The '775 patent is titled “Dynamic Batching for Inference System for Transformer-Based Generation Tasks” and was filed on December 3, 2021 as U.S. Patent Application No. 17/542,193 (“the '193 Application”). Ex. 1001, 1. The '193 Application is an original nonprovisional application. Ex. 1010, 5. The '193 Application was originally filed with claims 1-18. Ex. 1010, 51-56. There is no terminal disclaimer associated with the '775 patent. *See* Ex. 1001, cover; Ex. 2001, ¶38.

² Claim 10 recites similar substance in computer-readable medium format.

On February 1, 2022, Patent Owner filed an Information Disclosure Statement which disclosed the Gao reference (Ex. 1004). Ex. 1010, 110. On February 9, 2022, the Examiner rejected claims 1-4, 7-13, and 16-18 under 35 U.S.C. §103 as being unpatentable over Aarts et al. (US 2021/0192314 A1) in view of Chung et al. (US 10,846,096 B1). *Id.*, 120; Ex. 2001, ¶39.

The Examiner conducted an applicant interview on April 27, 2022, and the Aarts and Chung references were reconsidered. Ex. 1010, 187. Upon review, the Examiner agreed that both references apply specifically to RNNs and not to transformers. Ex. 1010, 189-191. Regarding Aarts, the Examiner stated, “Aarts also does not disclose batching *in the context of machine-learned transformer models*. In addition, Aarts also does not disclose applying a transformer model to a first batch and subsequently scheduling a second batch of requests...” (emphasis added). Ex. 1010, 191. As to Chung, the Examiner stated, “Chung does not describe batching *in the context of machine-learned transformer models*. In addition, Chung does not disclose applying a transformer model to a [first] batch of requests on a GPU...” (emphasis added). Ex. 1010, 189-190. The Examiner concluded that Aarts and Chung (which relate to RNNs) can be differentiated from the '193 Application (which relates to transformers), and therefore, “It was agreed that the proposed claim amendment... would overcome the current prior art rejections.” Ex. 1010, 177; *see* Ex. 2001, ¶40.

Patent Owner filed an Amendment to the claims that, *inter alia*, added text to claims 1 and 10: “wherein in a second set of inputs for the second batch of requests, a length of the sequence of input tokens for the new request is different from a length of an input for at least one request other than the new request.” Ex. 1010, 181. The Examiner allowed the ’193 Application on May 11, 2022. Ex. 1010, 195; *see* Ex. 2001, ¶41.

III. CLAIM CONSTRUCTION

The Board need not expressly construe any claim terms for the purpose of deciding whether to institute. In the related litigation, Petitioner asserted that one term requires construction and that two terms are indefinite under 35 U.S.C. §112(b). Ex. 2012. Patent Owner proposed that all terms be construed according to their plain and ordinary meaning, and that no terms are indefinite. *Id.* Patent Owner provided alternative constructions for each disputed term. *Id.* Claim construction briefing has been completed and the Markman hearing was held on October 29, 2024. Ex. 2013; Ex. 2014; *see also* Ex. 2001, ¶¶13-21.

Patent Owner agrees with Petitioner that the Board need not expressly construe any of the claim terms for the purpose of deciding whether to institute. Pet., 7-8. Yet Petitioner makes vague, unsupported statements in the Petition that Patent Owner interprets the claim terms with “unreasonable breadth,” without ever saying what construction it believes to be overbroad or what “narrower scope” it

deems appropriate. Pet., 7-8. Such hand waving does not rise to the level of claim construction.

If Petitioner seeks to assert a narrow construction of the claims, Patent Owner notes that narrowing claims from their plain and ordinary meaning – which meaning Petitioner admits Patent Owner asserted (Pet., 7) – is improper absent express disavowal of claim scope. *Retractable Techs., Inc. v. Becton, Dickinson & Co.*, 653 F.3d 1296, 1306 (Fed. Cir. 2011) (“To disavow claim scope, the specification must contain ‘expressions of manifest exclusion or restriction, representing a clear disavowal of claim scope.’”) (quoting *Epistar Corp. v. Int’l Trade Comm’n*, 566 F.3d 1321, 1335 (Fed. Cir. 2009)).

IV. STATE OF THE ART

A. Recurrent Neural Networks (RNNs)

In sequential data modeling, given a user prompt, a trained model iteratively generates tokens that represent text units (e.g., words, sub-words) as a response to the user prompt. Specifically, text units in the user prompt are converted into a sequence of input tokens. A token is a numerical representation of a text unit that allows machine learning (ML) and AI models to process the text units mathematically. Ex. 2001, ¶43. For example, if a token represents a word, a trained model processes the sequence of input tokens to generate an output token representing the likely next word in the sequence. Ex. 2004, pp. 137-138. The

output token is then used as the input to the model in the next iteration and the next output token is generated. This process is repeated until a token signaling the “end” is generated as the output token. Ex. 2001, ¶43; *see also id.*, ¶42.

RNNs are a class of neural network models that apply a trained set of parameters to input data in a recurrent manner to model and generate sequences of data (e.g., text, speech). The RNN includes connections that span adjacent iterations, introducing a notion of time to the model. Ex. 2002, §3. Although they date back to the 1980s, and the 1990s, use of RNNs became practical as the availability of digital data and computing power and memory increased. *See* Ex. 2009, 5-11, Ex. 2011, 182-185; *see* Ex. 2001, ¶44.

The RNN receives inputs (e.g., tokens from user prompt or previously generated output tokens) and generates the next output token. To generate the output tokens, RNNs generate a “hidden state” for each iteration that allows the model to maintain a memory state of the tokens that have been processed so far in the sequence when generating an output token for an iteration. The hidden state is iteratively updated across time, and acts as a compressed summary of the sequence processed so far. The length or dimensionality of the hidden state of an RNN is fixed over time. Ex. 2001, ¶45.

A representative RNN architecture is shown in FIG. A below, where the t -th input token is represented as $x^{(t)}$, the t -th hidden state is represented as $h^{(t)}$, and

the t -th output token is represented as $y^{(t)}$. Ex. 2002, §3; Ex. 2004, §7.3. While there are variations of RNNs, this description focuses on operations of the “Seq2Seq” RNN model, as this is the type of RNN model that is referred throughout the Petition. The Seq2Seq model includes an encoder RNN and a decoder RNN. Ex. 2001, ¶46.

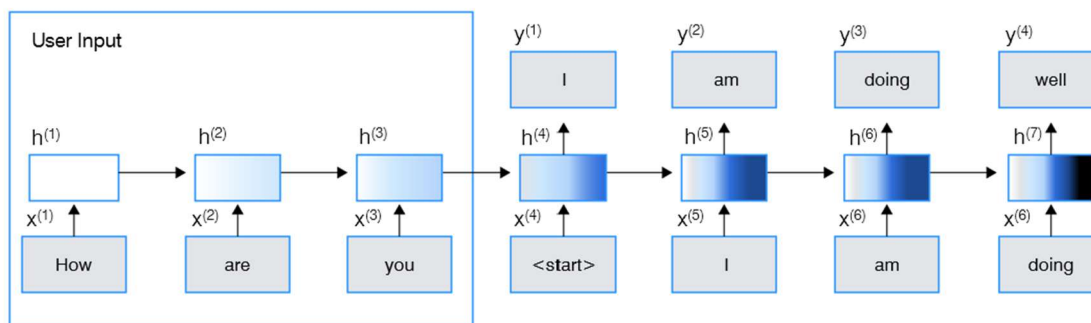


FIG. A. Overview of RNNs³

The encoder RNN during an “encoding phase” processes the tokens of the user prompt to generate a hidden state that incorporates the contextual information of the user prompt tokens that is later used to generate the first output token of the response. The hidden state $h^{(t)}$ is updated by combining the current input $x^{(t)}$ multiplied by a learned weight matrix and the previous hidden state $h^{(t-1)}$ multiplied

³ FIGS. A-F were created by Patent Owner to illustrate RNN and transformer models and batching in each.

by a learned weight matrix. Ex. 2002, §3; Ex. 2004, §7.3. For example, FIG. A illustrates the token for “How” is used to generate hidden state $h(1)$, the token for “are” and previous hidden state is used to generate hidden state $h(2)$, and the token for “you” and previous hidden state is used to generate hidden state $h(3)$. Ex. 2001, ¶¶46-47.

The decoder RNN during a “decoding phase” takes the final hidden state from the encoder RNN and iteratively updates the hidden state while iteratively generating output tokens. The output token at each iteration is generated based on the current hidden state $h^{(t)}$. FIG. A illustrates the start token “<start>” and the previous hidden state from the encoder RNN is used to generate hidden state $h(4)$, which is used to generate output token “I.” The previous output token “I” is used as the input for the next iteration and the previous hidden state is used to generate hidden state $h(5)$, which is used to generate output token “am.” The token “am” is used as the input for the next iteration and the previous hidden state is used to generate hidden state $h(6)$, which is used to generate output token “doing.” The token “doing” is used as the input for the next iteration and previous hidden state is used to generate hidden state $h(7)$, which is used to generate output token “well.” Ex. 2001, ¶48.

But RNNs suffer from inherent challenges. Ex. 2002, §3.2. As discussed above, the length of the hidden state of an RNN is fixed over time. Ex. 2005, §2. The performance of the RNN becomes limited as the sequence gets longer, since the

RNN layers compress the context of the sequence into the hidden state of fixed dimensionality. Ex. 2005, §2. This is shown in FIG. A via the diminishing size of the white and lighter portions within the hidden states over time. Ex. 2001, ¶49.

Another difficulty of RNNs is that RNNs require sequential processing, where each iteration during training depends on the output token of the previous iteration. Thus, it is difficult to parallelize computation, slowing down training especially for longer sequences. Moreover, the training process of the RNN makes it difficult to learn long-term dependencies because of the way the architecture of the RNN is designed. *See* Ex. 2002 (§3.2); Ex. 2001, ¶49.⁴

B. Transformers

Transformers are a newer class of revolutionary models for sequential data modeling that was developed in 2017, in part to address the shortcomings of RNNs. A key component of the transformer model is the attention mechanism. Ex. 2008, §2.4.1; *see generally* Ex. 2006. Given an input for a current iteration, the attention mechanism allows the transformer model to “attend” to different words of the sequence generated thus far when generating the next output token, so the

⁴ One technology that bridged RNNs into the use of transformers, but not discussed in the Petition, is LiSTM. *See* Ex. 2001, ¶50; Ex. 2017.

significance of specific words in the sequence are weighed when understanding the text. Ex. 2008, §2.3.1.; Ex. 2001, ¶51.

Transformer models model the contextual information for a sequence in an inherently different manner from RNNs. As tokens are generated for a sequence, the contextual information for each token is encoded as “keys” and “values” in each attention layer. For a given input at a current iteration (e.g., tokens for a user prompt or a previously generated output token), a “query” is generated at an attention layer that is a transformation of the input and embeds information on what the transformer model may want to pay attention to for generating the next output token. The attention layer also generates “keys” and “values” for each token in the sequence generated thus far. The “key” for a token in the sequence describes what the token is “offering” or when the token is important for the query. The “value” for a token in the sequence represents the content of that token within the generated context. Ex. 2001, ¶52.

The attention output for an attention layer is a weighted sum of the “values” for the tokens of the sequence, each weighted by a respective importance as determined by the respective “key” for the token with the “query” for the input. Therefore, different from RNNs, the transformer model for a given input allows the next output token to be generated by determining which tokens in the sequence are

important context for the given input and modeling the attention outputs as a summation of these tokens weighted by their importance. Ex. 2001, ¶53.

A representative transformer architecture is shown in FIG. B below. Given tokens for a user prompt, an attention layer during an “encoding phase” generates a key and a value for each input. The attention layer also generates a query for each input. For each query, the attention between the query and each of the keys for the respective tokens in the sequence are computed. The attention output for the query is computed as a weighted sum of the attention and the corresponding value for each token. Ex. 2001, ¶55.

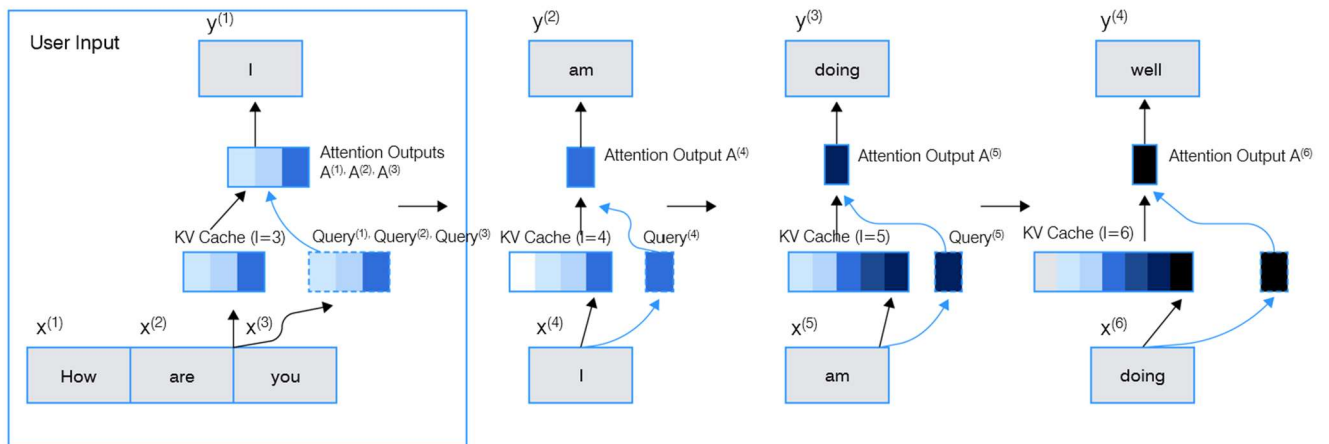


FIG. B. Overview of Transformer Model

For example, FIG. B illustrates tokens for a user prompt “How,” “are,” “you” that are initially input to an attention layer. The attention layer generates three keys

and three values corresponding to each token. The attention layer also generates three queries corresponding to each token. The attention outputs are generated and subsequent operations on the attention outputs are used to generate a prediction for the next token “I.” Ex. 2001, ¶¶55-56.

Due to the architecture of transformer models, the encoding phase of the transformer model typically can be performed in one forward pass iteration, whereas for the RNN model, the model is executed at least a number of iterations corresponding to the number of input tokens in the user prompt. The encoding phase of the transformer model therefore also reduces the number of iterations that have to be executed as compared to RNNs. This difference is highlighted in the portions of FIGs. A and B that are marked “user input,” where multiple iterations are performed for the encoding phase of the example RNN, and one iteration is performed for the encoding phase of the example transformer model. Ex. 2001, ¶57.

Moreover, different from the hidden state illustrated in FIG. A, transformer models service requests by maintaining a “KV Cache” for the attention layer that stores the “keys” (K) and the “values” (V) generated thus far for tokens in the sequence. Specifically, since the “keys” and “values” for a given token in the sequence will repeatedly be used to compute the attention outputs for subsequent iterations of inputs, the generated keys and values are stored in a “KV Cache” so

that the keys and values can be retrieved for subsequent iterations without having to recompute them. Ex. 2007, §2.1; Ex. 2001, ¶58.

The attention layer during one or more “decoding phases” iteratively generates output tokens until an end token <eos> (end-of-sequence) is generated. FIG. B illustrates at the next iteration, the previous output token “I” is used as the input and a key and a value for the input is generated and added to the KV Cache. The attention layer also generates a query for the input. The keys and values for the tokens are retrieved from the KV Cache. For the query, the attention between the query and each of the keys for the respective tokens in the KV Cache are computed. The attention output for the query is computed as a weighted sum of the attention and the corresponding value for each token in the KV Cache. The subsequent operations are applied to the attention output to generate the next token “am.” This process is repeated for tokens “doing,” and “well,” and at each iteration, the keys and the values are generated for the input are added to the KV Cache. Therefore, different from the hidden states of RNNs, the number of elements in the KV Cache for an attention layer proportionally grows with respect to the length of the sequence. Ex. 2001, ¶59.

Thus, a transformer model has an inherently different architecture from RNNs with respect to how sequential data is modeled through the attention layer. Specifically, as “keys” and “values” are generated for each token in a sequence, the

relative importance of each individual token can be assessed, rather than aggregating the context of the entire sequence in a hidden state of fixed length in RNNs. Ex. 2001, ¶60.

This difference results in many technical advantages that address the drawbacks of RNNs. Since the attention mechanism enables the transformer model to capture relationships between tokens regardless of their distance in the sequence, the model is adept at handling long-range dependencies. Ex. 2008, §2.3.2. This advantage is especially beneficial for complex tasks where context and relationships between distant elements are crucial for accurate interpretation and response. Ex. 2008, §2.3.2. Moreover, unlike RNNs that process sequential data step-by-step, transformer models can process tokens in parallel (e.g., such as processing input tokens in the encoding phase), leading to faster training times. Ex. 2008, §2.3.2. Ex. 2001, ¶61.

C. Batching in RNNs vs. Transformer Models

Servers (e.g., cloud services) often handle requests from a large number of devices and employ “batching” to increase parallelism and throughput. Ex. 2003, §1. When requests are processed separately, a significant amount of energy consumption, processing time, and communication cost is used to repeatedly access the learned parameter matrices of the trained model onto memory. Thus, during inference, batching merges multiple requests into a single batch so that the model

processes the requests together. This reduces the number of accesses for parameter matrices and allows the processing time and energy costs to be shared among the requests in the batch. Ex. 2003, §1; Ex. 2001, ¶62.

The operations in an RNN model that are batched requires lengths of the sequences to be the same across the batch of requests, i.e., the number of tokens in the user prompt and the number of iterations executed to be the same across different requests. Ex. 2003, §1. This is because during a batch operation, the data for the batch of requests are concatenated into a tensor, and the number of elements for a respective request in a tensor has to be the same across the requests for matrix multiplication to be performed efficiently. But same-length sequences are rarely the case in text modeling, as most requests have different user prompt lengths and different number of iterations until completion, both of which lead to variable sequence lengths. Ex. 2003, §1. Several methods were proposed to address these difficulties of RNN batching. Ex. 2001, ¶63.

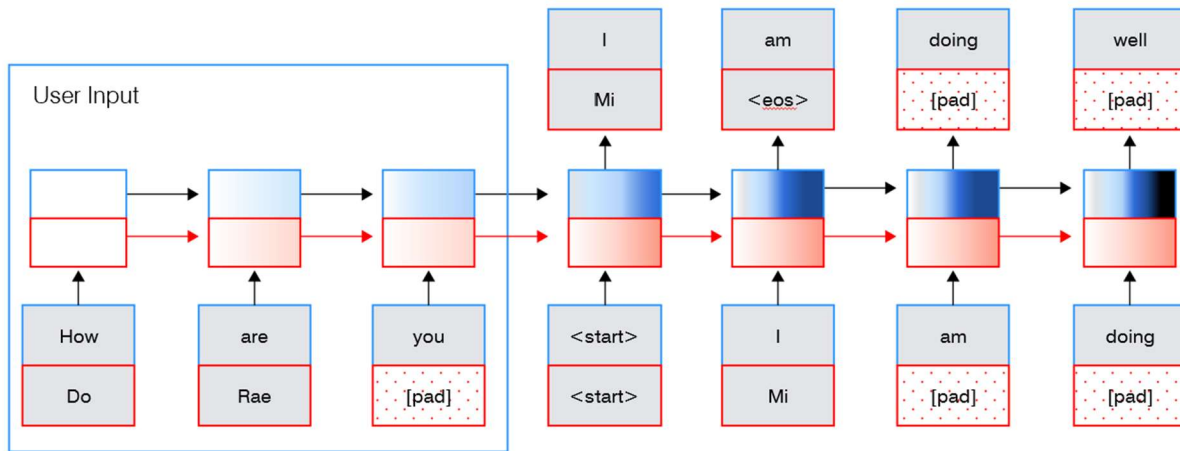


FIG. C. Batching in RNNs

One way was to perform “padding,” where padded tokens (e.g., zero tokens) are added to input tokens for the requests in a batch to match the length of the longest sequence in the batch. Ex. 2003, §1. FIG. C above illustrates an example of batching in RNNs for two requests. On top of the request in FIG. A, the example in FIG. C adds an additional second request with tokens for user prompt “Do,” “Rae.” During the encoding phase, since the second request only has two input tokens compared to three input tokens for the first request, a dummy token is inserted for the second request at the third iteration. Ex. 2001, ¶64.

During the decoding phase, after generating a first output token “Mi,” the second request is completed at the second decoding iteration as an end token <eos> is generated. But since the second request is batched together with the first request, the inputs and output tokens for the second request continue to be padded with

dummy tokens until the first request is completed. Therefore, a batch of requests start and complete together. Ex. 2001, ¶65.

Yet one main drawback of this batching is that the latency of all batched requests increases since the evaluation of requests cannot be completed until the longest sequence has been evaluated. Ex. 2003, §1. This results in underutilization of available parallel resources while waiting for the longest sequence to complete. Ex. 2003, §1; Ex. 2001, ¶66.

Another way to perform batching for RNNs is “cellular batching,” which focuses on batching requests at the granularity of cells or some iterations, instead of whole sequences. Ex. 2003, §2.2.3. Unlike padding, new requests are allowed to join a batch whose execution is ongoing. Ex. 2003, §2.2.3. Also, once a request is completed, the response can be returned to the user. Ex. 2003, §2.2.3; Ex. 2001, ¶67.

One reason why cellular batching is fairly feasible for RNNs is because the inputs to the RNN for the batch is typically a length of one token for each request, and the hidden states $h^{(t)}$ of the RNN model is of a fixed length regardless of the number of iterations. Therefore, batch operations that involve the hidden states could be performed by concatenating the hidden states of the batch of requests together regardless of whether the requests of the batch were executed at a different iteration number. Ex. 2001, ¶68.

Although transformer models were superior models in many aspects compared to RNNs, batching in transformer models remained a challenging problem. This is because for a batch of requests, the KV Cache for each request had to have the same lengths for batching to be performed, i.e., both the length of the user prompt tokens for a batch of requests and the number of iterations until the batch of requests were completed had to be the same. But this scenario rarely occurs since the KV Cache grows with the length of the sequence for transformer models, and different requests almost always generate a different number of output tokens until completion. While “padding” could be applied to transformer models to add padded tokens to the inputs and/or the KV Caches for the requests in a batch, it would result in similar latency and throughput drawbacks as described above. Ex. 2001, ¶69.

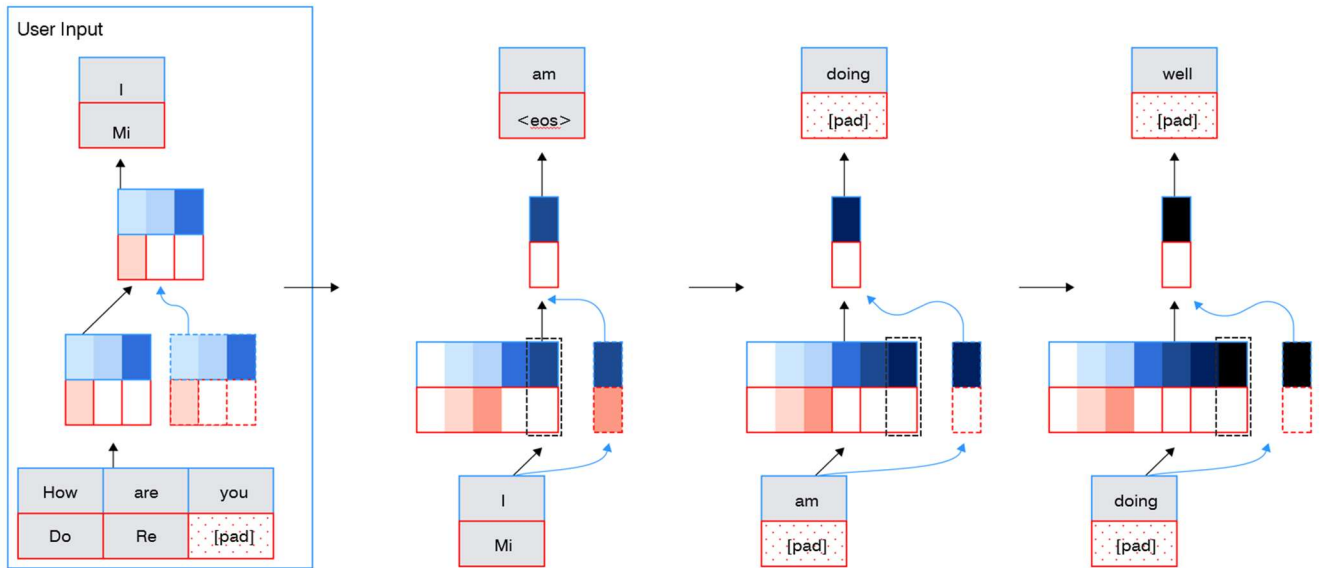


FIG. D. Static Batching in Transformer Models

FIG. D above illustrates static batching in transformer models. During the encoding phase, the input tokens for the second request are padded with one dummy token to match the length of three tokens for the first request. The first output tokens are generated for each of the requests. The batch of the first output tokens are input to the transformer model at the next iteration to generate second output tokens. The second request is completed since the end of sentence token <eos> is generated. However, since the requests are batched together, dummy tokens are input and generated as output tokens for the second request until the first request is completed.

Ex. 2001, ¶70.

D. Dynamic Batching for Transformer Models

The claimed invention solves the problem of conventional batching in transformers, as described above. FIG. E below illustrates dynamic batching in transformer models. Ex. 2001, ¶¶71.

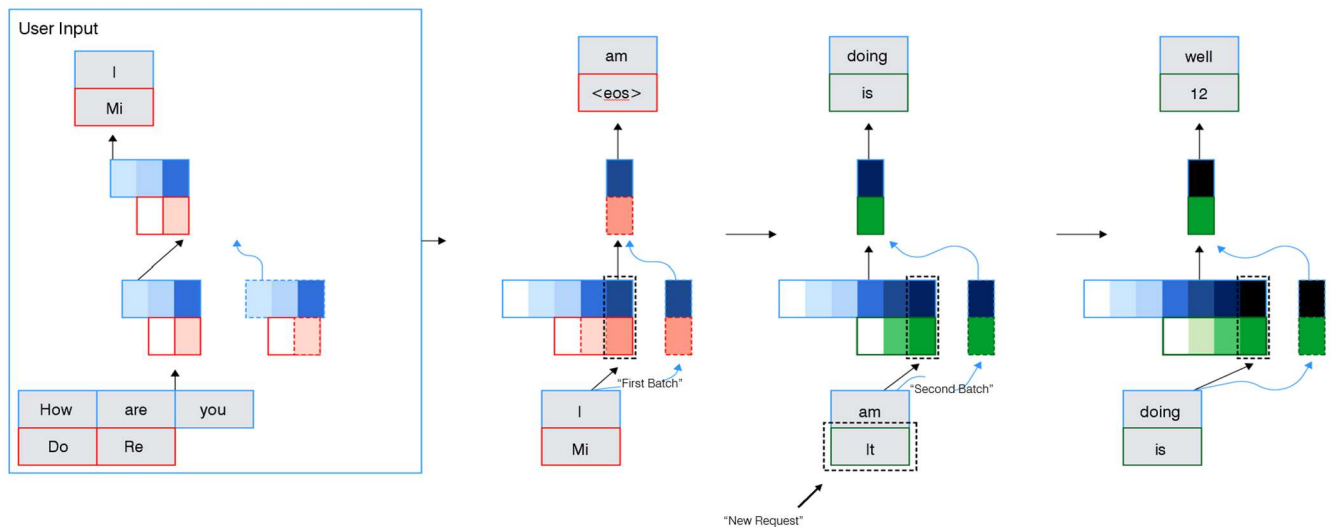


FIG. E. Dynamic Batching in Transformer Models

The batch of output tokens “I” for a first (blue outline) request and “Mi” for a second (red outline) request become a first set of inputs observed in the dynamic batching procedure after processing the user input (leftmost box). The transformer model is applied to the batch of requests thereafter to generate a set of output tokens, “am” for the first request (blue outline) and “<eos>” for the second request (red outline). The second request is completed since the end of sentence token <eos> is generated. Ex. 2001, ¶¶71-72.

Since the batch can be modified in dynamic batching, the second request is

removed and an input for a new request “It” is obtained. The batch of tokens “am” for the first request (blue outline) and “It” for the new request (green outline) become a new set of inputs for a second batch of requests that also include the new request. The transformer model is applied to the new batch of requests to generate the next set of output tokens, “doing” for the first request (blue outline) and “is” for the new request (green outline). This way, requests that are completed in a batch can be returned earlier to the user, and a new batch can also include a new request at an earlier point in time, reducing latency. Moreover, dummy tokens may not be needed since the length of the user prompt tokens or the length of the KV Cache are not required to be the same. Ex. 2001, ¶73.

V. OVERVIEW OF THE PRIOR ART

A. Katharopoulos

The Katharopoulos reference is titled “Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention,” by Katharopoulos et al., published in 2020 (“Katharopoulos”; “Ex. 1005”). This article title is misleading. Transformers, in fact, *are not* RNNs, and the Katharopoulos authors admit this. “In literature, *transformer models are considered to be a fundamentally different approach* to recurrent neural networks.” Ex. 1005, 5 (emphasis added); *see* Ex. 2001, ¶74.

Transformers represent a generational advance in neural network design over their RNN predecessors. Katharopoulos states that transformers, since their introduction by Vaswani, “have demonstrated impressive results” on tasks such as writing natural language or deciphering audio speech. Ex. 1005, 1. A downside to transformers is they “often come with a very high computational and memory costs.” *Id.*, 1. The reason for these high computational and memory costs are attributed to “self-attention, which processes contexts of N inputs with a quadratic memory and time complexity $O(N^2)$.” Ex. 1005, 1; *see* Ex. 2001, ¶75.

To address this problem, Katharopoulos proposes a technique called “linear transformers” that operate on a modified architecture. Ex. 1005, 2. These linear transformers perform similarly to a standard transformer but, it is claimed, are “up to 4000x faster on autoregressive prediction of very long sequences.” *Id.*, 1. The Katharopoulos linear transformer “significantly reduces the memory footprint and scales linearly with respect to the context length.” *Id.* Katharopoulos achieves this result by rewriting the transformer as an RNN. *Id.*, 2 (“in §3.4 we rewrite the transformer as a recurrent neural network.”); Ex. 2001, ¶76. Katharopoulos asserts the transformer “can be written as a model that, given an input, modifies an internal state and then predicts an output, namely a Recurrent Neural Network (RNN).” Ex. 1005, 5 (§3.4); *see* Ex. 2001, ¶76.

In the standard transformer model, the attention layer enables the transformer to perform its complex reasoning tasks. Ex. 2001, ¶77. Katharopoulos teaches “lineariz[ing] the attention computation,” or, in other words, handicapping the attention layer. Ex. 1005, 2. It would rewrite the transformer attention layer code from an “exponential kernel” into a simpler “linear” kernel. *Id.*, 3. The linear kernel then performs an attention calculation that generates *approximated* attention outputs. *Id.*, 2 (“Recent works ... have approximated softmax with a linear dot product... [W]e linearize the softmax attention in transformers”). Even Katharopoulos admits, “the feature function that corresponds to the exponential kernel is infinite dimensional, which makes *the linearization of exact softmax attention infeasible.*” *Id.*, 3 (§3.2) (emphasis added). The net effect is to improve computation speed but degrade output precision and quality. *Id.*, 2 (“we use the kernel formulation to speed up the calculation of self-attention”). Others in the field have pointed out the various drawbacks of Katharopoulos’s approach. Exs. 2018, 2019; *see also* 2015, 2016; *see* Ex. 2001, ¶77.

Since Katharopoulos essentially changes the attention mechanism in the transformer model, Katharopoulos does not maintain a KV Cache that stores the keys and values for individual tokens and grows in length as iterations are performed as described in Section IV.B. above. Neither does Katharopoulos compute the attention outputs in the way a transformer does. Ex. 2001, ¶78.

Katharopoulos teaches equation 20 as one of the core mathematical manipulations that is necessary to rewrite a transformer into an RNN:

$$y_i = f_l \left(\frac{\phi(x_i W_Q)^T s_i}{\phi(x_i W_Q)^T z_i} + x_i \right). \quad (20)$$

where the two types of hidden states for the converted RNN are given by:

$$s_i = s_{i-1} + \phi(x_i W_K) (x_i W_V)^T, \quad (18)$$

$$z_i = z_{i-1} + \phi(x_i W_K), \quad (19)$$

Ex. 1005, 5; Ex. 2001, ¶79. But Petitioner continues to describe the neural network after application of equation 20 as a “transformer model,” which it *is not*. The petition for *inter partes* review mistakenly states, “Katharopoulos teaches that a first set of output tokens (e.g., ‘ y_i , the i -th output’) by applying *the transformer model* (e.g., equation 20) to a first set of inputs...” Pet., 49 (emphasis added). To be clear, once Katharopoulos arrives at equation 20, all that remains is an RNN architecture. Ex. 1005, 5 (“The resulting RNN has two hidden states...”). The transformer-based neural network has been replaced. Ex. 2001, ¶80.

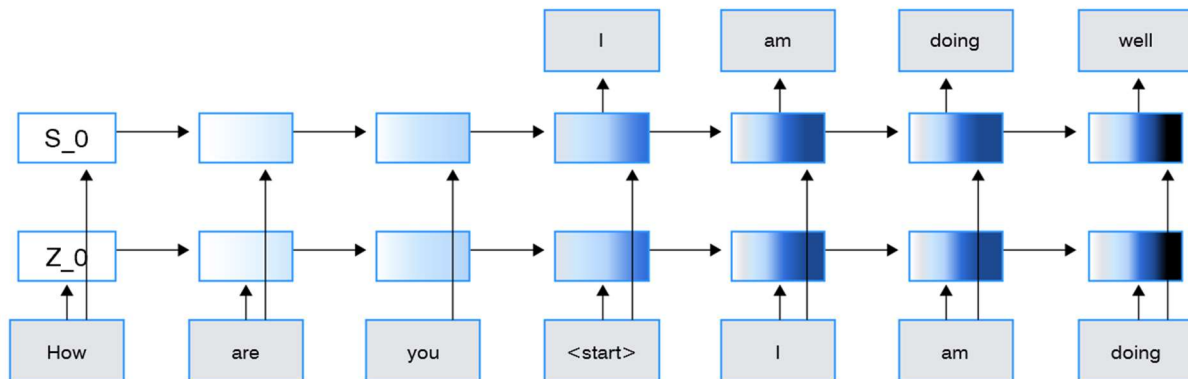


FIG. F. Example Representation of the Converted RNN in Katharopoulos

An example representation of the converted RNN in Katharopoulos is illustrated in FIG. F. As shown in the figure, the converted RNN is a variation of the RNN architecture shown in FIG. A that includes two hidden states, s_i and z_i . The two types of hidden states are generated over time and are used to generate the output tokens at each iteration. Ex. 2001, ¶¶80-81.

Katharopoulos converts a transformer model to an RNN as illustrated in FIG. F above, with two hidden states s_i and z_i . While these two hidden states are computed using queries, keys, and values, it does not compute the attention outputs in the way that transformer models do, neither does the converted RNN maintain a KV Cache for doing so. Therefore, the converted RNN in Katharopoulos would suffer from the same difficulties as explained with respect to RNNs, and no longer retains the architecture of a transformer model. Ex. 2001, ¶82; *See also id.*, ¶¶83-84.

B. Gao

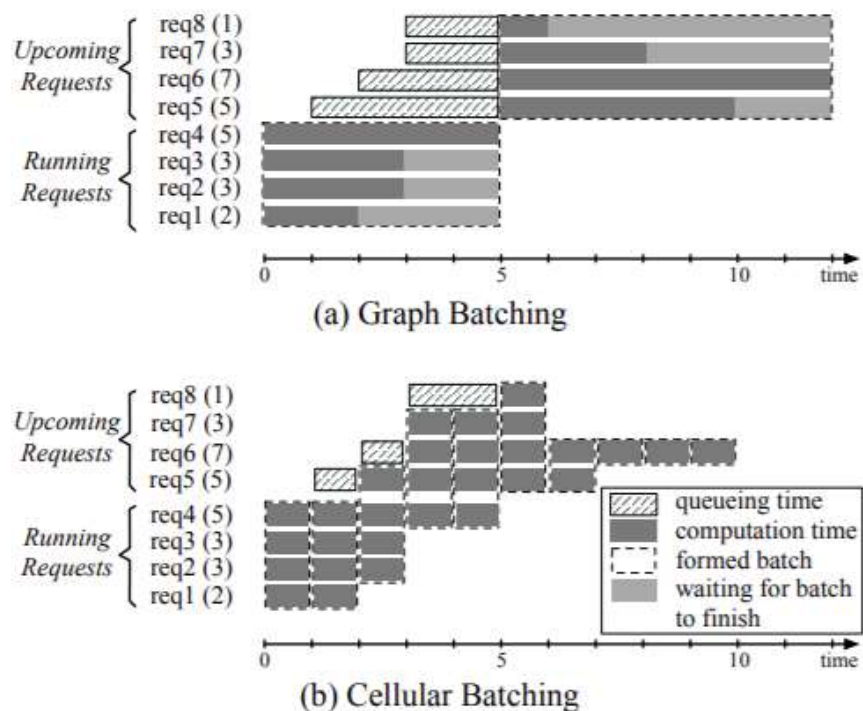
The Gao reference is titled “Low Latency RNN Inference with Cellular Batching” by Gao et al., published in 2018. Ex. 1004, 1. Gao proposes a technique called “cellular batching” to improve the latency and throughput of Recurrent Neural Network (RNN) inference. *Id.*; Ex. 2001, ¶85.

In layman's terms, batching is the process of organizing a large pile of data into chunks of a predetermined size before feeding it into a neural network. *See* Pet., 19-20. Different batching techniques are commonly used to take advantage of different neural network architectures and hardware platforms. Pet., 21. When data inputs are uniform in length (i.e., they have the same “computation graph”), batching is a straightforward process. Ex. 1004, 3. This is the case for batching in the Multi-layer Perceptron (MLP) model or in the Convolution Neural Network (CNN) model. *Id.*; Ex. 2001, ¶86.

The common RNN batching techniques, referred to as “graph batching,” are inefficient. Ex. 1004, 4 (“Graph batching incurs extra latency due to unnecessary synchronization because an input cannot start executing unless all requests in the current batch have finished.”). While the RNN architecture “is designed to model *variable length inputs*,” graph batching techniques work best with “inputs [of] *the same length*.” *Id.*, 1 (emphasis added). Thus, in practice, graph batching in RNNs can incur delays, and these delays are exacerbated “when inputs have varying

lengths, causing some long input to delay the completion of the entire batch.” *Id.*, 4. Each variable-length input can have a different “recursion depth.” *Id.*, 3. As a result, the application of graph batching to RNNs presents a practical problem that “negatively affects both the latency and throughput.” *Id.*, 4; Ex. 2001, ¶87.

Gao addresses this problem with a solution called “cellular batching.” Ex. 1004, 4. Gao’s key insight is to segment the recursive RNN data flow into many equally-sized cells, or “computation units,” which can be drawn as a sequence of cells in an organism. *Id.*, 2. Thus, Gao enables an RNN to perform batching at a smaller “cell” granularity instead of a larger “graph” granularity. *Id.* Gao illustrates this concept in Figure 5:



Id., 4. Ex. 2001, ¶88.

As Gao admits, the cellular batching solution is RNN-specific, and “cellular batching is *only beneficial for RNN inference*.” *Id.*, 14 (emphasis added). Although the Gao reference was published in 2018 and transformers were invented in 2017, Gao never suggests applying cellular batching to transformers. Gao never once mentions transformer models. Ex. 2001, ¶89.

VI. PETITIONER FAILED TO SHOW IT IS MORE LIKELY THAN NOT THAT THE CLAIMS ARE INVALID UNDER 35 U.S.C. §103

A. Petitioner Has Not Met Its Statutory Burden

In an *inter partes* review instituted under this chapter, the *petitioner shall have the burden* of proving a proposition of unpatentability by a preponderance of the evidence.” 35 U.S.C. §316(e) (emphasis added). An IPR petition may be considered *only if* “the petition *identifies*, in writing and *with particularity*, each claim challenged, the grounds on which the challenge to each claim is based, and the evidence that supports the grounds for the challenge to each claim.” 35 U.S.C. §312(a)(3) (emphasis added).

Petitioner has not met that burden at least because the Petition and the references both suffer from fatal deficiencies. The Petition is rife with erroneous citations and thus fails to set out the evidence with the required specificity, the Gao and Katharopoulos references are disqualified as non-analogous art, and the Katharopoulos reference was not established as being publicly available by the

critical date. Each of these defects alone are sufficient to deny institution, even before reaching the substantive deficiencies of the references as applied.

1. Citation Errors Foreclose Required “Particularity” Showing

The Petition is replete with citation errors such that it does not “identif[y]...with particularity... the evidence that supports the grounds for the challenge to each claim” as required by 35 U.S.C. §312(a)(3). In particular, the substance of the Petition has at least 15 erroneous citations, which cite to either the wrong reference or the wrong page of the cited reference, including at least one erroneous citation in the evidence supporting each of limitations 1[A], 1[C], and 1[F] of independent claim 1.⁵ In addition, there are multiple misquotes of the cited evidence. Through these errors, “Petitioner has placed the burden on the Board to ascertain how the prior art allegedly reads on the challenged claims—a task [the Board should] not undertake.” *Corning Incorp. v. DDSM IP Assets B.V.*, IPR2013-00043, Paper 95 at 13 (P.T.A.B. May 1, 2014), quoting *DeSilva v. DiLeonardi*, 181 F.3d 865, 866-67 (7th Cir. 1999) (“A brief must make all arguments accessible to the judges, rather than ask them to play archaeologist with the record.”); *see also*

⁵ Incorrect cites on pages 2, 23, 24, 25, 26 (2), 41 (1[A]), 48 (2)(1[C]), 57 (1[F]), 64, 69, 75, 77, and 78.

Amazon Web Servs., Inc. et al. v. Saint Regis Mohawk Tribe, IPR2019-00103, Paper 22 at 15 (P.T.A.B. May 10, 2019) (declining to “play archaeologist with the record”); *A.C. Dispensing Equip. Inc. v. Prince Castle LLC*, IPR2014-00511, Paper 16 at 5-6 (P.T.A.B. Sep. 10, 2014) (“Petitioner should not expect the Board to search the record and piece together the evidence necessary to support Petitioner’s arguments”); *Purdue Pharma L.P. v. Depomed, Inc.*, IPR2014-00377, Paper 17 at 5-6 (P.T.A.B. Aug. 6, 2014) (“It is not the Board’s role to play archeologist to uncover any additional support in the record that is not raised and discussed in the Petition and that may bolster [the expert’s] opinion.”). The Petition should be denied on this basis alone.

2. The Petition Improperly Relies On Non-analogous Art

Gao and Katharopoulos do not qualify as prior art under §103 because they are non-analogous art. *Innovation Toys, LLC, v. MGA Ent., Inc.*, 637 F.3d 1314, 1321 (Fed. Cir. 2011) (holding a reference qualifies as prior art for an obviousness determination under §103 only when it is analogous to the claimed invention). “Two separate tests define the scope of analogous prior art: (1) whether the art is *from the same field of endeavor*, regardless of the problem addressed and, (2) if the reference is *not within the field of the inventor's endeavor*, whether the reference still is *reasonably pertinent to the particular problem* with which the inventor is involved.” *In re Bigio*, 381 F.3d 1320, 1325 (Fed. Cir. 2004) (emphasis added);

see also In re Klein, 647 F.3d 1343, 1348 (Fed. Cir. 2011). Gao and Katharopoulos are not analogous to the claimed invention under either test.

The invention of the '775 patent “relates generally to machine-learning *transformer* neural network models, and more particularly to selective batching *for transformer models*.” Ex. 1001, 1:7-9 (emphasis added). The claims specifically recite methods and computer program products executing batches of requests using “a machine-learning *transformer* model.” *See* Ex. 1001, 27:55-30:47 (emphasis added). The specification distinguishes transformer models from “other recurrent machine-learning models (e.g., RNNs)” that use state data that have the same lengths across a batch of requests. Ex. 1001, 24:28-38. Neither Gao nor Katharopoulos pertain to batching for transformer models. Ex. 2001, ¶90.

The field of endeavor test requires a determination of the appropriate field of endeavor by reference to explanations of the invention's subject matter *in the patent application*, including the embodiments, function, and structure of the claimed invention. *Bigio*, 381 F.3d at 1325; *see also In re Wood*, 599 F.2d 1032, 1036 (C.C.P.A. 1979) (confining the field of endeavor to the scope explicitly specified in the background of the invention). It is clear from the specification that the appropriate field of endeavor of the '775 patent is batching *for transformer models*. Ex. 2001, ¶¶90-91.

Neither Gao nor Katharopoulos are within this field of endeavor. Gao's field of endeavor is batching on RNNs, rather than transformer models. Ex. 1004, 1 (Abstract), 14 (Conclusion). Katharopoulos's field of endeavor is converting transformers into RNNs. Ex. 1005, 2. None of the disclosure of Gao or Katharopoulos is about transformers or even analogous to the embodiments, function, and structure of the claimed invention under the field of endeavor test. Ex. 2001, ¶91.

Under the second test, "[a] reference is reasonably pertinent if, even though it may be in a different field from that of the inventor's endeavor, it is one which, because of the matter with which it deals, logically would have commended itself to an inventor's attention in considering his problem." *In re Clay*, 966 F.2d 656, 659 (Fed. Cir. 1992). "If a reference disclosure *has the same purpose* as the claimed invention, the reference relates to the same problem, and that fact supports use of that reference in an obviousness rejection," whereas "[i]f [a reference] is directed to *a different purpose*, the inventor would accordingly have had less motivation or occasion to consider it." *Id.* (emphases added).

The purpose of the '775 patent is to solve the problem of how to *process a batch of requests with variable lengths in transformer models* without using more resources. Ex. 1001, 1:48-51 (emphasis added); *see also id.*, Abstract. Gao's purpose is to improve both the latency and throughput of RNN inference, via cellular

batching. Ex. 1004, 1 (Abstract). This is a different purpose than that of the '775 patent, because RNNs have a fixed length hidden state. Katharopoulos's purpose is to speed up autoregressive inference of certain transformers by "rewrit[ing] the transformer as a recurrent neural network." Ex. 1005, 2; *see also id.*, 1. Thus Katharopoulos also is directed to a different purpose than the '775 patent. Applying each test, neither Gao nor Katharopoulos are analogous art. Ex. 2001, ¶92.

3. The Petition Did Not Establish Public Availability of the Katharopoulos Reference

Petitioner has not met its burden with respect to Katharopoulos to "identify, with particularity, *evidence sufficient to establish a reasonable likelihood that the reference was publicly accessible before the critical date* of the challenged patent and therefore that there is a reasonable likelihood that it qualifies as a printed publication." *Hulu, LLC, v. Sound View Innovations, LLC*, IPR2018-01039, Paper 29 at 12–13 (P.T.A.B. Dec. 20, 2019) (precedential) (emphasis added). The Petition must "identify...the prior art relied upon and *evidence that it qualifies as such.*" *Id.*

Petitioner bears the burden to prove that the asserted references qualify as printed publications. *Acceleration Bay, LLC v. Activision Blizzard, Inc.*, 908 F.3d 765, 772 (Fed. Cir. 2018). In particular, "[a] reference is considered publicly accessible if it was 'disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising

reasonable diligence, can locate it.”” *Id.* The Petition provided no evidence whatsoever that Katharopoulos (Ex. 1005) was either publicly disseminated or publicly available at the relevant time. Petitioner identified no relevant dates or indicia of dissemination. Indeed, Petitioner ***did not even argue*** that Katharopoulos was made publicly available for the conference. Nor did Dr. Hammouri provide any testimony to that effect. *See* Ex. 1002, ¶¶76-82.

The Petition states that Katharopoulos was “published and presented” in 2020 but provides no evidence of publication. Pet., 25. The Board has found that submission of a similar conference paper did not show a reasonable likelihood that it was published before the critical date. *See Meta Platforms, et al. v. VL Collective IP LLC*, IPR2023-00924, Paper 11 at 48-49 (P.T.A.B. Dec. 6, 2023); *see also Dynaenergetics US, Inc. v. Geodynamics, Inc.*, PGR2018-00065, Paper 8 at 14–19 (P.T.A.B. Nov. 15, 2018) (Board holding that a conclusory public availability assertion coupled with a citation to a supporting declaration was insufficient to meet a petitioner’s burden); *In-Depth Geophysical, Inc. v. Conocophillips Co.*, IPR2019-00849, Paper 14 at 10-11 (P.T.A.B. Sept. 6, 2019) (finding that a date on the face of the document and copyright date “sheds virtually no light on whether the document was publicly accessible prior to” the critical date).

Nor does anything on the face of the document provide definitive indicia of publication or dissemination. The face of the document bearing a 2020 copyright is

not sufficient. *See ServiceNow Inc., v. Hewlett-Packard Co.*, IPR2015-00716, Paper 13 at 8, 15-17 (P.T.A.B. Aug. 26, 2015) (finding copyright date on face of document insufficient absent a hearsay exception or exclusion). The face of Katharopoulos also bears an “arXiv” date stamp of 2020. Ex. 1005, 1. But arXiv appears to be a preprint repository. *See* List of preprint repositories (row 6), Wikipedia, *available at* https://en.wikipedia.org/wiki/List_of_preprint_repositories#cite_note-3.

Submission to a preprint repository or peer review committee in connection with submitting a paper to a conference suggests an absence of actual public accessibility. *SRI Int’l, Inc. v. Internet Sec. Sys.*, 511 F.3d 1186, 1197 (Fed. Cir. 2008).

Petitioner failed to meet its burden to establish that Katharopoulos constitutes prior art. The Board has denied institution based on similar circumstances. *First Solar, Inc. v. Rovshan Sade*, IPR2023-00827, Paper 13 at 18 (P.T.A.B. Nov. 16, 2023) (petitioner not having offered evidence, or even argument, that a reference webpage was indexed or otherwise locatable by a search engine). The same result should obtain here.

B. Defects in Obviousness Analysis

A claim is unpatentable under 35 U.S.C. §103(a) if “the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.”

KSR Int'l Co. v. Teleflex Inc., 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations, including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) objective evidence of nonobviousness, i.e., secondary considerations. *See Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 17 (1966).

“[I]t is error to reach a conclusion of obviousness until all [the *Graham*] factors are considered.” *Apple Inc. v. Samsung Elecs. Co.*, 839 F.3d 1034, 1048 (Fed. Cir. 2016) (en banc) (citations omitted). “This requirement is in recognition of the fact that each of the *Graham* factors helps inform the ultimate obviousness determination.” *Id.* “In an [*inter partes* review], the petitioner has the burden from the onset to show with particularity why the patent it challenges is unpatentable.” *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016) (citing 35 U.S.C. §312(a)(3) (requiring *inter partes* review petitions to identify “with particularity . . . the evidence that supports the grounds for the challenge to each claim”)). This burden of persuasion never shifts to Patent Owner. *See Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015) (discussing the burden of proof in *inter partes* review). Furthermore, Petitioner cannot satisfy its burden of proving obviousness by employing “mere conclusory statements.” *In re Magnum Oil Tools Int'l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016).

1. The Proposed Combination Suffers from Fatal Flaws for Failure to Show Certain Claim Elements

The proposed combination fails to establish a *prima facie* case of obviousness because it does not disclose or suggest several claimed limitations. Ex. 2001, ¶93. If even a single element of the claim is absent from the prior art, the claim cannot be considered obvious. *See CFMT, Inc. v. YieldUp Int'l Corp.*, 349 F.3d 1333, 1342; *see also Garmin Int'l, Inc. v. Patent of Cuozzo Speed Techs. LLC*, IPR2012-00001, Paper 15 at 15 (P.T.A.B. Jan. 9, 2013) (refusing to institute an *inter partes* review under 35 U.S.C. §103 where prior art did not disclose all claim limitations); *In re Rijckaert*, 9 F.3d 1531, 1534 (Fed. Cir. 1993) (reversing obviousness rejection where prior art did not teach or suggest all claim limitations).

a. The proposed combination does not meet the claim limitations that explicitly recite “running,” “access[ing],” and “applying” a transformer model (1[pre], 1[a], 1[c], 1[f]; 10[pre], 10[a], 10[c], 10[f])

Claim 1 recites use of a transformer model throughout:

1[pre] A method of dynamically executing batches of requests on one or more execution engines **running a machine-learning transformer model**, comprising:

1[a] receiving, by a serving system, one or more requests for execution, the serving system including a scheduler and one or more execution engines each coupled **to access a**

machine-learning transformer model including at least a set of decoders;

...

1[c] generating, by the execution engine, a first set of output tokens by **applying the transformer model** to a first set of inputs for the batch of requests, wherein **applying the transformer model** comprises applying at least one batch operation to one or more input tensors associated with the batch of requests;

...

1[f] generating, by the execution engine, a second set of output tokens by **applying the transformer model** to the second set of inputs for the second batch.

Ex. 1001, 27:55-28:20 (claim 1); *see also id.*, 29:1-36 (similar limitations of claim 10);⁶ Ex. 2001, ¶94.

As discussed *supra* in Section II.A., the very purpose of the '775 patent is to solve the problem of how to process a batch of requests with variable lengths *in transformer models* without using more resources. Ex. 1001, 1:48-51. Each of

⁶ The arguments regarding claim 1 apply equally to claim 10. And although the dependent claims are not addressed herein, they inherit these deficiencies from the independent claims on which they depend. Ex. 2001, ¶26.

the four limitations above explicitly recite operations occurring in a transformer model. The transformer model is not optional. Ex. 2001, ¶95.

Gao and Katharopoulos, alone or in combination, do not disclose or suggest batching *for transformer models*. Rather, the Petition's proposed combination converts a transformer model to an RNN (Katharopoulos), and then applies "cellular batching" to the resulting RNNs (Gao). See Ex. 1005, 2; Ex. 1004, 1; see also Ex. 2001, ¶96. The Petition admits that Gao relates to RNNs and not a transformer model and that Katharopoulos "represents" the transformer model as an RNN. Pet., 40. Katharopoulos reformulates the "attention layer" of a transformer to convert the transformer model into an RNN. Ex. 1005, 5. At that point, *there is no longer a transformer model*. In the proposed combination, Gao is applied to the resultant RNN. Pet., 44-49. Therefore, if Katharopoulos was combined with Gao as the Petition proposes, the resultant combination at most would disclose iteration-level batching with RNNs, not transformers as claimed. Ex. 2001, ¶96.

(1) 1[pre]

Petitioner fails to show that the references render this limitation obvious. The Petition asserts that 1[pre] is disclosed by Gao, and argues that "a POSITA would have been motivated to combine Gao's teaching of dynamically executing batches of requests on one or more execution engines with Katharopoulos's teaching of a running a machine-learning transformer model." Pet., 38. The Petition does not

explain how Gao's batches would be modified to execute on Katharopoulos's transformer model. *Id.* And as discussed in Section V.A. *supra*, Katharopoulos does not disclose a ***transformer model***. Ex. 2001, ¶97.

(2) 1[a]

This limitation includes the requirement “one or more execution engines each coupled to access a machine-learning transformer model.” The Petition relies on Gao's “workers” as the execution engines. Pet., 39. The Petition then discusses Katharopoulos's functionality, and then makes the conclusory statement that “Gao's Workers (i.e., execution engines) each have access to a machine-learning transformer model [of Katharopoulos] including at least a set of decoders.” Pet., 41. Petitioner never says ***how*** Gao's workers would “each ***[be] coupled to*** access [the] machine-learning transformer model” of Katharopoulos, as required by the claims. Again, Katharopoulos does not disclose a ***transformer model***. See Section V.A.; Ex. 2001, ¶98.

(3) 1[c]

Petitioner fails to show that the references render this limitation obvious. The Petition relies on the combination as follows: “applying the transformer model (*e.g.*, equation 20) [of Katharopoulos] to a first set of inputs (*e.g.*, ‘ x_i denotes the i -th input’) for the batch of requests, ***wherein applying the transformer model comprises applying at least one batch operation, as disclosed by Gao***, to one or more input

tensors associated with the batch of requests.” Pet., 49 (emphasis added). This combination is problematic because the limitation specifically recites applying *the transformer model* by *applying at least one batch operation*. As noted above, Gao’s batch operation is disclosed as being performed on RNNs, which is not “applying the transformer model,” as claimed. Ex. 2001, ¶¶99-100. Moreover, once Katharopoulos arrives at equation 20, all that remains is an RNN architecture. Ex. 1005, 5. The transformer-based neural network has been replaced, so there is no “transformer model” present in formula 20. Ex. 2001, ¶100.

(4) 1[f]

For this limitation, the Petition merely repeats its arguments from limitation 1[c]. Pet., 54-57. Thus, Petitioner fails to show that the references render limitation 1[f] obvious due to the same deficiencies noted for limitation 1[c].

In addition, limitation 1[f] is not merely “a second iteration of the first ‘generating’ limitation” (1[c]) as the Petition suggests. Recall that Petitioner relied on Katharopoulos for the “first set of inputs (*e.g.*, ‘ x_i denotes the i -th input’).” Pet., 49. However, for the “second set of inputs” in 1[e], the Petition relies on a “second batch of requests” from Gao. Pet., 52. Proper antecedent basis requires that “*the* second set of inputs” of limitation 1[f] are the same ones recited in limitation 1[e]. The Petition fails to explain how the “second batch of requests” from Gao

could be *inputs* into formula 20 of Katharopoulos. Ex. 2001, ¶101; *see also id.*, ¶¶99-100.

b. The combined references do not disclose or suggest “the second batch of requests scheduled responsive to determining that the execution engine has memory available to execute the second batch of requests” (1[e])

Gao and Katharopoulos, alone or in combination, do not disclose or suggest this limitation. As discussed *supra* in Section IV.B., transformer models maintain a KV Cache that stores the keys and values (i.e., context) of tokens in the sequence, and which grows proportionally as the length of the sequence increases over time. Therefore, claim limitation 1[e] schedules the second batch of requests responsive to determining that the execution engine has memory available to execute the second batch of requests. Since the amount of memory for storing the KV Cache for a request can grow over time and require more memory, the claim limitation first determines whether the execution engine has memory available to execute the second batch of requests before executing the second batch of requests. *See* Section IV.B; Ex. 2001, ¶102.

Neither Gao nor Katharopoulos teaches *determining whether an execution engine has memory available* for executing a second batch of requests. The Petition incorrectly asserts that “it would be obvious to a POSITA to first ensure that memory is available before execution” (Pet., 52) and “Gao’s batches have allocated memory

depending on their size so the number of requests in a batch is a direct reflection of allocated memory availability.” Pet., 53; *see* Ex. 2001, ¶103.

But the proposed combination **would not need** a memory check, since the proposed combination at most teaches batching in RNNs, which have a **fixed hidden state length** across iterations. In particular, the “converted RNN” of Katharopoulos includes hidden states that are denoted as s_i and z_i , and that each have a fixed length across iterations of i . Ex. 1005, 5 (§3.4). Therefore, the proposed combination does not disclose or suggest determining that an execution engine has memory available to execute a second batch of requests. Ex. 2001, ¶103.

c. The Petition Fails to Establish a *Prima Facie* Case of Obviousness

Petitioner's analysis lacks one of the three threshold inquiries **required** by the obviousness analysis: ascertaining the differences between the prior art and the claims at issue. *Graham*, 383 U.S. at 17-18. The Petition never identifies any differences between the references and the claims. Since it does not consider these differences, Petitioner has not met its burden to establish a *prima facie* case. *See Apple v. Samsung*, 839 F.3d at 1048 (“[I]t is error to reach a conclusion of obviousness until all [the *Graham*] factors are considered.”). Institution should be denied.

2. Petitioner Fails to Prove a Motivation to Combine

To establish obviousness, there must be “a reason” for a skilled artisan “to combine the elements *in the way the claimed new invention does.*” *KSR Int’l*, 550 U.S. at 418(emphasis added). Motivation to combine is a factual determination. *Novartis AG v. Torrent Pharms. Ltd.*, 853 F.3d 1316 (Fed. Cir. 2017). Petitioner has the burden to explain what changes are necessary to the references and *how* the substitution would meet the claim limitations. *See Gamber-Johnson LLC v. Nat’l Products Inc.*, IPR2021-01159, Paper 7 at 38-39 (P.T.A.B. Jan. 3, 2022) (finding lack of motivation where petitioner did not make this showing).

There are clear flaws in Petitioner’s claimed motivation to combine. To start, no POSITA with the intention of improving transformer performance would select Gao and Katharopoulos. Nor would a POSITA rewrite a state-of-the-art transformer model into an older, inferior RNN. Petitioner offers no rationale for why a POSITA would make such a conversion. Even if a POSITA did combine these references, the combination would eliminate the transformer model entirely and defeat the principle of operation. In addition, the resultant RNN model would require retraining, and Petitioner has not explained how this retraining might be accomplished. The only remaining rationale for the combination is hindsight. Thus, Petitioner’s combination must fail.

a. A POSITA would not select and combine these references as proposed.

The obviousness analysis is not whether aspects of prior art references *could be* combined, but whether a POSITA *would* select and combine them when trying to solve the problem of the '775 patent. “[A] patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art.” *KSR Int’l*, 550 U.S. at 418.

Each of Petitioner’s proffered rationales for combining the references start with the assumption that a POSITA would have selected and combined these references. For example, Petitioner argues that a POSITA would have been motivated to combine the references because “[i]t is straight forward” to do so and because the technology of one reference “can be leveraged” to be used in the other reference. Pet., 27-28. This showing is not sufficient. Rather, “obviousness requires the additional showing that a person of ordinary skill at the time of the invention *would have selected and combined* those prior art elements” to arrive at the claimed invention. *Unigene Labs., Inc. v. Apotex, Inc.*, 655 F.3d 1352, 1360 (Fed. Cir. 2011) (emphasis added). Here, a POSITA *would not*. Ex. 2001, ¶¶104-106.

(1) A POSITA faced with the problem of processing a batch of requests with variable lengths in transformers would not select the Gao and Katharopoulos references

A POSITA would not have selected the Gao and Katharopoulos references when faced with the problem the inventors sought to solve. The '775 patent states that it sought to solve the problem of the potential infeasibility of processing a batch of requests with variable lengths in transformers. Ex. 1001, 1:46-51. As discussed above, Gao and Katharopoulos each are both from different fields of endeavor *and* serve a different purpose. *See supra* Section VI.A.2. A POSITA would not select—or even consider—references directed instead to rewriting transformers to linear transformers or RNNs (Katharopoulos) and performing cellular batching in RNNs (Gao). A POSITA simply would not consider these references reasonably pertinent to the problem to be solved. Petitioner provided no rationale whatsoever to the contrary to establish the “*require[d] additional showing* that a person of ordinary skill at the time of the invention *would have selected and combined* those prior art elements.” *Unigene Labs.*, 655 F.3d at 1360 (emphasis added). Ex. 2001, ¶107.

(2) A POSITA would not have combined references in a way that eliminates the transformer model entirely

The proposed combination improperly defeats the principle of operation of the patent by writing out use of a transformer model entirely. MPEP §2143.01, citing

In re Ratti, 270 F.2d 810, 813 (C.C.P.A. 1959) (“If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious.”)

Katharopoulos takes transformers and approximates them as RNNs – thereby eliminating the transformer model entirely. Specifically, Katharopoulos describes modifying the transformer layer to rewrite it as RNN. Ex. 1005, 5. Katharopoulos acknowledges the transformer is replaced at equation 20 shown below: “In the following equations, we formalize the transformer layer of equation 1 as a recurrent neural network. The resulting RNN has two hidden states...”

$$y_i = f_l \left(\frac{\phi(x_i W_Q)^T s_i}{\phi(x_i W_Q)^T z_i} + x_i \right). \quad (20)$$

Ex. 1005, 5. Once that rewriting occurs, there is no longer any transformer model.

Ex. 2001, ¶108. The combination then applies cellular batching to the resultant RNN, as described in Gao. *See* Ex. 1004, 4. Wholesale rewriting of one neural network model (transformer) to result in a different model (RNN) improperly defeats the principle of operation and thus cannot prove obviousness. *Ratti*, 270 F.2d at 813; *see* Ex. 2001, ¶¶108-109.

(3) A POSITA would not be motivated to modify a state-of-the-art transformer model to approximate to a less sophisticated RNN model

To go from a transformer model to an RNN model is to go backwards in time in the development and sophistication of machine learning models. A POSITA would not be motivated to do so. Ex. 2001, ¶110.

RNNs were invented in the 1980's. Meanwhile, transformers have revolutionized AI research since their invention in 2017. *See* Ex. 1009, 1. As discussed *supra* in Section IV.B., transformers were in fact developed to address some of the shortcomings of RNNs. For example, to avoid recursion to allow parallel computation, to eliminate vanishing and exploding gradients, and to reduce drops in performance due to long dependencies. Transformer-based neural networks perform far better than older RNN models at complex tasks including pattern recognition, evaluating context, and making predictions. Thus, to rewrite a transformer as an RNN model is essentially to handicap the transformer model, leading to an inferior result. In practice, no POSITA would do so. Ex. 2001, ¶110.

Moreover, as described above with respect to FIG. B, transformer models allow the encoding phase to be performed in one iteration, as the tokens for the user prompt can be processed together to generate the first output token, providing additional improvement in latency and computational efficiency. Since the proposed modification with Gao and Katharopoulos is a converted RNN, the proposed

modification would require the encoding phase to be performed for a number of iterations that correspond to the number of tokens in the user prompt, resulting in diminished latency compared to transformer models. Ex. 2001, ¶111.

(4) A POSITA would not be motivated to modify the references as proposed

The proposed combination modifies the disclosure of the references to read in two features that are not taught or suggested by either reference. Rather, Petitioner fabricates these modifications out of whole cloth. Ex. 2001, ¶112.

First, the references do not disclose or suggest mapping of the “Decoder layer” in Katharopoulos to “a single RNN cell.” Pet., 33. The Petition alleges that “[i]n the work of Katharopoulos, each Decoder layer is viewed as a single RNN cell (Decoder-Cell).” *Id.* It is not clear based on the Petition or the Expert Declaration what “a cell” of an RNN is or corresponds to. Even if there were a definition, Katharopoulos never mentions a “cell”; thus it is unclear how a POSITA would arrive at mapping a “Decoder layer” in Katharopoulos to a “RNN cell” of Gao. Petitioner never explains why a POSITA would be motivated to make this proposed modification. Indeed, Petitioner does *not even acknowledge* that the combination requires modification of the references. Ex. 2001, ¶113.

Second, the references do not disclose or suggest generation of a $(L \times n)$ “computational graph.” Pet., 34-35. The Petition alleges that a $(L \times n)$

“computational graph” of RNN cells can be created based on the combination of Katharopoulos and Gao that would enable cellular batching to be performed on batches of requests: “The computational graph . . . can be viewed as a matrix of Decoder-Cells with size (L x n). Each layer . . . comprises identical cells that process the different encoded input elements and only differ in the state-vector.” *Id.* But Katharopoulos appears to discuss the conversion of an attention mechanism to an RNN for only *a single* Decoder layer. It is unclear how a POSITA would arrive at a computational graph comprising multiple layers (or rows) of RNN cells from the teachings of the references. Again, Petitioner does *not even acknowledge* that the combination requires modification of the references, much less explain why a POSITA would be motivated to make the proposed modification. Ex. 2001, ¶114.

Thus, Petitioner fails to explain how the references could be entirely reconfigured such that (1) the “Decoder layer” in Katharopoulos could be mapped to “a single RNN cell and (2) they result in generation of a (L x n) “computational graph.” These modifications are unsupported by the disclosure of the references and are not articulated with a reasoned analysis in the Petition or by Dr. Hammouri. *See Magnum Oil*, 829 F.3d at 1380.

b. Petitioner provides no valid rationale for the combination

Petitioner's purported rationales supporting its combination fail because they fail to consider the disadvantages of the combination, rely on unsupported rationales and parroted expert testimony, and are the result of improper hindsight.

(1) Petitioner's rationale neglects to consider the disadvantages of the combination

When considering a modification or combination of art, a POSITA would consider both the advantages and disadvantages that result. *See Winner Int'l Royalty Corp. v. Wang*, 202 F.3d 1340, 1349 n.8 (Fed. Cir. 2000) (“[T]he benefits, both lost and gained, should be weighed against one another.”); *see also Merck & Cie v. Gnosis S.p.A.*, 808 F.3d 829, 834 (Fed. Cir. 2015) (the prior art must be considered for all its teachings rather than selectively). Petitioner points to the combination as “improv[ing] the latency and throughput” of the neural network and producing “significant speed ups.” Pet., 21, 31. Yet Petitioner is silent on the drawbacks of the combination. Ex. 2001, ¶115.

RNNs have several disadvantages as compared to transformers. For example, any “speed ups” made by the proposed combination would be offset by performance reductions in other areas, such as the neural network's attention, context, and predictive ability. Ex. 2001, ¶116. RNNs perform much worse in terms of memory loss and at complex tasks including pattern recognition, evaluating context, and

making predictions. *Id.* In addition, the RNN model that would result from the combination would need to be retrained, which is necessarily slow because RNNs require sequential processing, where each iteration during training depends on the output token of the previous iteration. *Id.*

Petitioner never acknowledges this need, never explains how this retraining would be accomplished on the resultant RNN, and never accounts for the added expenditure of time and resources. *See Gamber-Johnson LLC*, IPR2021-01159, Paper 7 at 38-39 (rejecting argument that a POSITA would have been motivated to substitute one part with another where petitioner failed to explain what other corresponding changes were necessary and how the substitution would meet the claim limitations). Indeed, Katharopoulos acknowledges that RNNs are less efficient to train than transformers. Ex. 1005, 4 (§3.3.2). Petitioner ignores the effect of these disadvantages of RNNs. Ex. 2001, ¶117.

(2) Petitioner relies on fictional, unsupported rationales

Petitioner manufactures a rationale for the combination out of whole cloth. Petitioner never cites to the references themselves in the motivation to combine section, makes unexplained leaps of logic in its reasoning, and even misrepresents the record in an effort to support its fictional motivation to select and combine the references.

Importantly, Petitioner does not cite to Katharopoulos or Gao even once in the more than 10 pages it dedicates to its motivation to combine analysis. *See* Pet., 27-37. Indeed, for much of its analysis, it is unclear whether Petitioner is relying on the disclosure of Gao, Katharopoulos, or neither due to the lack of citations. *See, e.g.,* Pet., 32-33. Ex. 2001, ¶118.

Petitioner also makes unexplained leaps of logic, improperly leaving Patent Owner and the Board to try to piece together the arguments. *See DeSilva v. DiLeonardi*, 181 F.3d at 866-67 (“A brief must make all arguments accessible to the judges, rather than ask them to play archaeologist with the record.”); *see also supra* Section II (cases cautioning against making the Board do so). For example, Petitioner asserts that each transformer layer (“decoder block”) in the figure is mapped to a corresponding RNN cell. Pet., 30. But Katharopoulos never uses the word “cell” And the Petition does not describe what in Katharopoulos would be mapped to a “cell” in Gao, and since Katharopoulos only describes one attention layer and how a POSITA would connect the other attention layers (1, 2, . . . , L) in a way that could be batched in Gao. Ex. 2001, ¶119.

As another example, Petitioner's rationale requires logical leaps that do not follow from the disclosure of the references. For example, Petitioner relies on Katharopoulos as teaching that transformers can be rewritten as RNNs, and then batching in RNNs as taught by Gao. Petitioner then argues that the RNN batching

of Gao can be modified so it applies to the transformers of Katharopoulos. But doing so would seem to require that Katharopoulos rewrite RNNs as transformers, which is the opposite of the rewriting that it teaches. The Petition does not explain how Gao's batching would otherwise apply to the linearized transformers of Katharopoulos that are its starting point *before* it rewrites them as RNNs. Ex. 2001, ¶120.

Petitioner also relies on its lack of citations to misrepresent the record. For example, Petitioner states that Katharopoulos discloses that “even a softmax attention-function could be cast as an RNN cell” (Pet., 31; no citation to Katharopoulos), but leaves out that Katharopoulos explicitly states that “linearization of exact softmax attention [is] *infeasible*.” Ex. 1005, 3 (§3.2.1) (emphasis added). This framing is misleading. Ex. 2001, ¶121.

As a result, Petitioner provides no valid rationale for its combination.

(3) Petitioner's expert's parroted testimony cannot save its faulty analysis

No evidence supports Petitioner's rationale for combining the references. As noted above, Petitioner's motivation to combine analysis contains zero citations to either Gao or Katharopoulos. The Petition cites to 28 paragraphs of the declaration of Dr. Hammouri, Petitioner's expert, in this section. Pet., 27-37, citing Ex. 1002, ¶¶83-110. However, these paragraphs are a *nearly verbatim* parroting of the Petition

itself, differing by only a handful of words out of over 2,500. Nor does Dr. Hammouri provide evidence to support his opinions about what a POSITA would “understand” from the references. Ex. 2001, ¶¶122-123. Thus, Dr. Hammouri's expert testimony is conclusory and cannot establish a motivation to combine. *See Magnum Oil*, 829 F.3d at 1380 (finding conclusory statements do not satisfy petitioner's burden to establish obviousness); *SFC Co. v. LG Chem Ltd.*, IPR2020-00178, 2020 WL 2790521, at *13 (P.T.A.B. May 29, 2020) (denying institution where petitioner's alleged motivation to combine relied solely on conclusory statements).

Moreover, Dr. Hammouri's unsupported expert testimony is entitled to little or no weight, as discussed further below. *See Rohm & Haas Co. v. Brotech Corp.*, 127 F.3d 1089, 1092 (Fed. Cir. 1997); *Xerox Corp. v. Bytemark, Inc.*, IPR2022-00624, Paper 12 at 5 (P.T.A.B. Feb. 10, 2023); *see infra* Section VI.C.

(4) Hindsight is the only possible rationale to combine Gao and Katharopoulos

Faced with a lack of evidence to support its motivation to combine, no consideration of the drawbacks of the proposed combination, and references a POSITA would not select or combine, improper hindsight bias is all that is left to explain Petitioner's unlikely selection and combination of Gao and Katharopoulos. At any rate, Petitioner's attempt to Frankenstein together the prior art to result in the

claims, based only on the teachings of the '775 patent itself, should be rejected. Such hindsight reasoning that “us[es] the invention as a roadmap to find its prior art components” is insufficient to establish a motivation to combine. *See Ruiz v. A.B. Chance Co.*, 357 F.3d 1270, 1275 (Fed. Cir. 2004); *see also Metalcraft of Mayville, Inc. v. The Toro Co.*, 848 F.3d 1358, 1367 (Fed. Cir. 2017) (“[W]e cannot allow hindsight bias to be the thread that stitches together prior art patches into something that is the claimed invention.”). Petitioner’s lack of evidence or rationale beyond broad, conclusory assertions are insufficient. *See ActiveVideo Networks, Inc. v. Verizon Commc’ns, Inc.*, 694 F.3d 1312, 1327 (Fed. Cir. 2012) (expert’s conclusory statements that POSITA would know how to combine references held “fraught with hindsight bias”). Ex. 2001, ¶124.

C. Dr. Hammouri’s Testimony Should Be Afforded Little Weight

Petitioner’s expert testimony is conclusory and does not save the faulty analysis of the petition. *See Xerox Corp.*, IPR2022-00624, Paper 12 at 5. The vast majority of Dr. Hammouri’s expert testimony relied upon in the Petition is unsupported and consists of self-serving, ***conclusory assertions*** without any underlying factual support. Indeed, Dr. Hammouri largely parrots attorney argument verbatim from the petition. *E.g., compare* Ex. 1002, ¶¶120-134 *with* Pet., 38-45. His opinions therefore should be given little weight. Ex. 2001, ¶125.

In particular, Petitioner relies on Dr. Hammouri's testimony to support why missing limitations would be obvious to POSITA. But Dr. Hammouri provides *no supporting data* to back up his statements. E.g., compare Ex. 1002, ¶¶83-111 with Pet., 27-38; see Ex. 2001, ¶122. In *Xerox*, Director Vidal specified that "little weight" should be given to such expert testimony, since it "merely repeats, verbatim, the conclusory assertion for which it is offered to support." *Xerox*, IPR2022-00624, Paper 12 at 3-4; see also *Rohm & Haas*, 127 F.3d at 1092 (unsupported expert testimony is entitled to little or no weight); *ActiveVideo Networks*, 694 F.3d at 1327 (expert's conclusory statements that POSITA would know how to combine references held "fraught with hindsight bias"). Dr. Hammouri's expert testimony should be given little weight. Ex. 2001, ¶126.

VII. CONCLUSION

For the reasons provided above, the Petition requesting *inter partes* review of claims 1-18 of the '775 Patent fails to show it is more likely than not that any of the challenged claims are unpatentable. Therefore, the Petition should be denied and the Board should decline to institute an *inter partes* review of the '775 Patent.

IPR2024-01234
Patent Owner's Preliminary Response

Dated: November 15, 2024

FENWICK & WEST LLP

/Jennifer R. Bush/
JENNIFER R. BUSH
Reg. No. 50,784
Attorney for Patent Owner

**CERTIFICATE OF COMPLIANCE
WITH TYPE-VOLUME LIMITATION**

Pursuant to 37 C.F.R. §42.14 *et seq.*, the undersigned certifies that this Preliminary Response complies with the 14,000-word type-volume limitation. The Preliminary Response contains 13,215 words, excluding the parts of the Preliminary Response exempted from the type-volume limitation.

Dated: November 15, 2024

FENWICK & WEST LLP

/Jennifer R. Bush/

JENNIFER R. BUSH

Reg. No. 50,784

Attorney for Patent Owner

CERTIFICATION OF SERVICE

The undersigned certifies that the foregoing PATENT OWNER'S PRELIMINARY RESPONSE TO PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT 11,442,775 and accompanying Exhibits 2001-2009 and 2011-2019 were served on Petitioner's lead and backup counsel in their entirety by electronic service at the following email addresses provided below:

James P. Murphy
Polsinelli PC
1000 Louisiana Street, Suite 6400
Houston, TX 77002

jpmurphy@polsinelli.com

Adam P. Daniels
Polsinelli LLP
2049 Century Park E., Suite 2900
Los Angeles, CA 90067

adaniels@polsinelli.com

Dated: November 15, 2024

FENWICK & WEST LLP

Fenwick & West LLP
801 California Street
Mountain View, CA 94041

/Jennifer R. Bush/
JENNIFER R. BUSH
Reg. No. 50,784
Attorney for Patent Owner