IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

FRIENDLIAI INC.,	)	
	)	C.A. No. 23-816-MN
Plaintiff,	)	
	)	**JURY TRIAL DEMANDED**
v.	)	
	)	**PUBLIC VERSION**
HUGGING FACE, INC.,	)	
	)	
Defendant.	)	

JOINT CLAIM CONSTRUCTION BRIEF

David E. Moore (#3983)
Bindu A. Palapura (#5370)
Andrew L. Brown (#6766)
POTTER ANDERSON & CORROON LLP
Hercules Plaza, 6th Floor
1313 N. Market Street
Wilmington, DE 19801
Tel: (302) 984-6000
dmoore@potteranderson.com
bpalapura@potteranderson.com
abrown@potteranderson.com

OF COUNSEL:

Michael J. Sacksteder
Shreyas A. Kale
Samantha Ong
FENWICK & WEST LLP
555 California Street, 12th Floor
San Francisco, CA 94104
Tel: (415) 875-2300

Jessica Kaempf
FENWICK & WEST LLP
401 Union Street, 5th Floor
Seattle, WA 98101
Tel: (206) 389-4550

Attorneys for Plaintiff FriendliAI Inc.

Stephen J. Kraftschik (#5623)
POLSINELLI PC 222 Delaware Avenue,
Suite 1101
Wilmington, DE 19801
(302) 252-0920
skraftschik@polsinelli.com

OF COUNSEL:

Jason A. Wietjes
POLSINELLI PC
2950 N. Harwood St., Ste. 2100
Dallas, TX 75201
(214) 661-5519

Adam P. Daniels
POLSINELLI LLP
2049 Century Park E.
Los Angeles, CA 90405, Ste. 2900
(310) 556-6754

Jahnathan L.D. Braquet
POLSINELLI PC
1000 Louisiana Street, Suite 6400
Houston, TX 77002
(713) 374-1600

Attorneys for Defendant Hugging Face, Inc.

Dated: October 11, 2024
11765408 / 23295.00001
Public Version Dated: October 17, 2024

TABLE OF CONTENTS

		Page
I.	Introduction	1
II.	The Asserted Patents	2
III.	Level of Ordinary Skill in the art	13
IV.	Legal FRAMEWORK	13
	A. Claim Terms are Presumed to Have Their Plain and Ordinary Meanings	13
	B. Indefiniteness Must be Proven by Clear and Convincing Evidence	14
	A. Claim Construction	15
	B. Indefiniteness	16
V.	Agreed-Upon Claim Terms	16
VI.	Disputed Claim Terms	17
	A. "dedicated cache memory"/ "cache memory dedicated"	17
	1. Plaintiff's Opening Position	17
	2. Defendant's Answering Position	21
	3. Plaintiff's Reply Position	25
	4. Defendant's Surreply Position	29
	B. "scheduler"	31
	1. Plaintiff's Opening Position	32
	2. Defendant's Answering Position	34
	3. Plaintiff's Reply Position	36
	4. Defendant's Surreply Position	38
	C. "at least one batch operation"	39
	1. Plaintiff's Opening Position	40
	2. Defendant's Answering Position	41
	3. Plaintiff's Reply Position	46

4. Defendant's Surreply Position 51

VII. CONCLUSION 54

TABLE OF AUTHORITIES

Page(s)

Cases

3M Innovative Properties Co. v. Tredegar Corp.,
725 F.3d 1315 (Fed. Cir. 2013).....27

Augme Techs., Inc. v. Yahoo! Inc.,
755 F.3d 1326 (Fed. Cir. 2014).....37

Baran v. Med. Device Technologies, Inc.,
616 F.3d 1309 (Fed. Cir. 2010).....46

Bicon, Inc. v. Straumann Co.,
441 F.3d 945 (Fed. Cir. 2006).....21, 29

Biosig Instruments, Inc. v. Nautilus, Inc.,
783 F.3d 1374 (Fed. Cir. 2015).....14

Chef Am., Inc. v. Lamb–Weston, Inc.,
358 F.3d 1371 (Fed. Cir. 2004).....17, 32

Fenner Investments, Ltd. v. Cellco P'ship,
778 F.3d 1320 (Fed. Cir. 2015).....52

GE Lighting Sols., LLC v. AgiLight, Inc.,
750 F.3d 1304 (Fed. Cir. 2014).....19, 26

Hill-Rom Servs., Inc. v. Stryker Corp.,
755 F.3d 1367 (Fed. Cir. 2014).....14

Interval Licensing LLC v. AOL, Inc.,
766 F.3d 1364 (Fed. Cir. 2014).....39

Ironburg Inventions Ltd. v. Valve Corp.,
64 F.4th 1274 (Fed. Cir. 2023) 34, 37

Johnson Worldwide Assocs., Inc. v. Zebco Corp.,
175 F.3d 985 (Fed. Cir. 1999).....19, 27

Markman v. Westview Instruments, Inc.,
52 F.3d 967,980 (Fed. Cir. 1995) (en banc), *aff'd*, 517 U.S. 370 (1996) 52

Markman v. Westview Instruments, Inc.,
52 F.3d 967 (Fed. Cir. 1995), aff'd, 517 U.S. 370 (1996).....15

Mentor H/S, Inc. v. Med. Device Alliance, Inc.,
244 F.3d 1365 (Fed. Cir. 2001).....17, 33

Merck & Co. v. Teva Pharm. USA, Inc.,
395 F.3d 1364 (Fed. Cir. 2005).....14

Meyer Intellectual Properties Ltd. v. Bodum, Inc.,
690 F.3d 1354 (Fed. Cir. 2012).....30

Modine Mfg. Co. v. U.S. Int'l Trade Comm'n,
75 F.3d 1545 (Fed. Cir. 1996).....15

Nautilus, Inc. v. Biosig Instruments, Inc.,
572 U.S. 898 (2014).....14, 16, 38, 45

Phillips v. AWH,
415 F.3 d 1303, 1318 (Fed. Cir. 2005) (en banc).....30

Phillips v. AWH Corp.,
415 F.3d 1303 (Fed. Cir. 2005) (en banc)..... *passim*

PPC Broadband, Inc. v. Corning Optical Communications RF, LLC,
815 F.3d 747 (Fed. Cir. 2016).....46

Promptu Sys. Corp. v. Comcast Corp.,
92 F.4th 1372 (Fed. Cir. 2024) 21

SentriLock, LLC v. Carrier Fire & Sec. Americas LLC,
C.A. No. 20-520 (MN), 2024 WL 3328495 (D. Del. July 8, 2024) 38

SoftView LLC v. Apple Inc.,
2013 WL 4758195 (D. Del. Sept. 4, 2013).....15

Sprint Communications Co. L.P. v. Cox Communications Inc.,
302 F. Supp. 3d 597 (D. Del. 2017).....54

Teva Pharm. USA, Inc. v. Sandoz, Inc.,
789 F.3d 1335 (Fed. Cir. 2015).....15

Teva Pharms. USA, Inc. v. Sandoz, Inc.,
574 U.S. 318 (2015).....13, 15

TIP Systems, LLC v. Phillips & Brooks/Gladwin, Inc.,
529 F.3d 1364 (Fed. Cir. 2008).....46

TQ Delta, LLC v. 2Wire, Inc.,
373 F. Supp. 3d 509 (D. Del. 2019).....37

Vitronics Corp. v. Conceptronic, Inc.,
90 F.3d 1576 (Fed. Cir. 1996).....14

Wyers v. Master Lock Co.,
616 F.3d 1231 (Fed. Cir. 2010).....30

Statutes

35 U.S.C. § 112(b).....32, 40

TABLE OF EXHIBITS

Exhibit No.	Description
1	U.S. Patent No. 11,836,520 ('520 Patent)
2	U.S. Patent No. 11,442,775 ('775 Patent)
3	Declaration of Michael Pazzani
4	Newton's Telecom Dictionary (30th ed. 2016)
5	A Dictionary of Computing, Oxford University Press (6th ed. 2008)
6	Dictionary of Computing, A&C Black Publishers (6th ed. 2010)
7	Prosecution History of U.S. Patent No. 11,442,775
8	Prosecution History of U.S. Patent No. 11,836,520
9	The American Heritage Dictionary (4th ed. 2006)
10	Encyclopedia Britannica, *cache memory*, May 20, 2021, https://www.britannica.com/technology/cache-memory (last visited Sep. 5, 2024)
11	The Authoritative Dictionary of IEEE Standard Terms (7th ed. 2000)
12	A Dictionary of Computer Science (7th ed. 2016)
13	Videotaped Deposition Transcript of Nicolas Patry, Dated July 25, 2024

I. INTRODUCTION

Plaintiff's Opening Introduction

Pursuant to the Court's Scheduling Order (D.I. 107), plaintiff FriendliAI Inc. ("FriendliAI") submits its opening brief regarding construction of the disputed claim terms identified in the parties' Revised Joint Claim Construction Chart (D.I. 109).

FriendliAI asserts two patents in this action—U.S. Patent No. 11,442,775 ("the '775 Patent") and U.S. Patent No. 11,836,520 ("the '520 Patent") (collectively, the "Asserted Patents")—against defendant Hugging Face, Inc. ("Hugging Face" or "Defendant"). The parties dispute the meaning of three claim terms across the two patents-in-suit, two of which Hugging Face contends are indefinite.

Under Federal Circuit precedent, where the meaning of the patent claim term is clear, no construction is necessary. The terms and phrases at issue here—namely, "dedicated cache memory," "scheduler," and "at least one batch operation"—are readily understood from the claim language and specification, and therefore do not require construction. To the extent the Court finds that the terms should be construed, FriendliAI has provided alternate constructions that are consistent with each claim term's plain and ordinary meaning in light of the claim language and specification. By contrast, Defendant proposes constructions that ignore the plain and ordinary meaning of the claim terms. Defendant proposes that two of the claim terms ("scheduler" and "at least one batch operation") are indefinite, and proposes to construe the remaining claim term ("dedicated cache memory") in a manner that inserts extraneous limitations that are absent from the claims and specification. The Court should reject Defendant's proposals and construe the disputed terms according to their plain and ordinary meanings.

Defendant's Answering Introduction

Pursuant to the parties' stipulation and order to extend time (D.I. 107) Defendant

Hugging Face, Inc. (Hugging Face) submits this brief in response to Plaintiff FriendliAI, Inc.'s (FriendliAI) opening claim construction brief.

Plaintiff's Reply Introduction

Defendant's proposed claim construction for "dedicated cache memory"/ "cache memory dedicated" violates fundamental claim construction canons, and its indefiniteness challenges to the claim terms "scheduler" and "at least one batch operation" are unsupported by any expert opinion, rely on fundamental misunderstandings of the patent specifications, and otherwise fall far short of showing indefiniteness by clear and convincing evidence. The Court should reject Defendant's proposed construction and indefiniteness arguments, and construe the terms according to their plain and ordinary meanings.

Defendant's Surreply Introduction

Hugging Face submits its sur-reply claim construction brief as set forth herein. In summary of its prior briefing, Hugging Face requests that the Court adopt its proposed construction for the disputed "dedicated cache memory" term and find the disputed "scheduler" and "at least one batch operation" terms indefinite.

II. THE ASSERTED PATENTS

Plaintiff's Opening Asserted Patents

The '775 and '520 Patents, entitled "Dynamic Batching for Inference System for Transformer-Based Generation Tasks," share a common specification. Both patents are directed to a system and method for dynamically batching inference requests for transformer models. (Ex. 2 at Abstract.) A transformer model is a neural network architecture used in artificial intelligence, introduced in the paper "Attention Is All You Need." ("Attention Is All You Need," Vaswani et al., *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing*

Systems (2017).) Transformer models are executed by inference systems to process requests from a user. (Ex. 2 at 1:26-30.) For example, a transformer model may receive a query from a user then will generate a response to the query, or it may receive from a user a paragraph of text in one language then will generate a translation or summary of the text. (*Id.* at 1:14-25.)

Transformer models offer several improvements over pre-existing technologies. Because the transformer model architecture leverages a self-attention mechanism, it is particularly well-suited for developing and deploying Large Language Models (LLMs). (Ex. 3 at ¶¶ 26-28.) One significant advantage is that transformers overcome prior limitations of relying only on nearby words to create the context for understanding words with multiple meanings. *Id.* Transformer models excel at processing sequences of data, such as text or speech, and transforming them into another sequence, for example, translating text into another language, answering questions, or summarizing information. *Id.* They are a key factor in the success of systems such as the popular ChatGPT, which are trained on vast amounts of data to perform tasks such as answering questions, summarizing, or translating text. *Id.*

But while transformer models offer improvements over the prior art, executing such models previously encountered latency and throughput challenges, particularly in large-scale applications. Specifically, while inference systems can process requests in batches (i.e., process multiple requests in a batch together) to maximize processor utilization (Ex. 1 at 1:38-43), batching for transformer models was often not feasible. That is because transformer models involve requests that are variable in length, whereas batching methods often require the length of data for multiple requests in a batch to be the same in order to be processed. (*Id.* at 1:46-51, 1:55-61, 2:50-57; *see also id.* at 10:34-37.)

The invention of the ’775 and ’520 Patents overcomes such limitations and provides an

optimization technique known as batching with iteration-level scheduling, also known as dynamic batching or iteration-level dynamic batching. This novel invention allows for batching of transformer models, which significantly reduces latency and improves throughout in executing transformer-based requests.

An embodiment of the patented invention is depicted in the following figure:

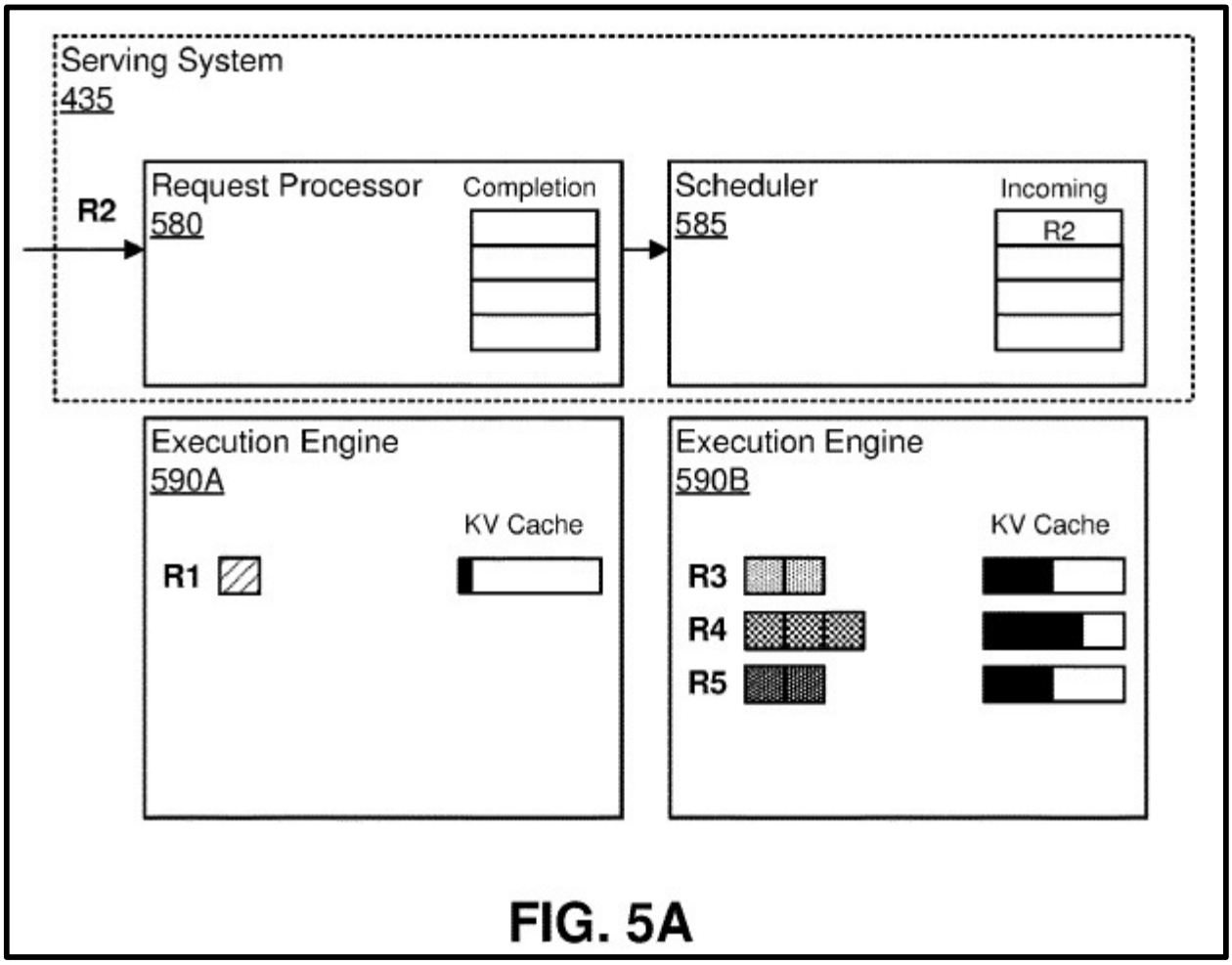


FIG. 5A

(Ex. 1, Fig. 5A; Ex. 2, Fig. 5A.) As shown above, the serving system 435 is comprised of the request processor 580 and the scheduler 585. (Ex. 1 at 22:32-35; Ex. 3 at ¶ 29.) A user may submit requests to the serving system, which are depicted in Fig. 5A by the labels R1-R5. (Ex. 1 at 22:35-45, 22:54-59; Ex. 3 at ¶ 29.) Requests are received by the serving system and sent to the scheduler. (*Id.*)_ The scheduler then adds those requests to batches of requests, and schedules them for

execution on an execution engine. (Ex. 1 at 22:45-51; Ex. 3 at ¶ 29.)_ When memory becomes available, new requests may also be added to a batch to be executed with one or more of the previously executing requests. (Ex. 1 at 23:32-35; Ex. 3 at ¶ 29.) By enabling batching for transformer models, this innovation substantially lowers the requirements for hardware, electricity, and cooling to execute transformer-based requests, thereby significantly reducing the cost of deploying LLMs.

Defendant's Answering Asserted Patents

A. Relevant Teachings

The '775 patent and the '520 patent share a common specification.[1] The asserted patents describe an inference system that applies a machine-learning transformer model to batches of input requests with variable input lengths. Ex. 2 at Abstract. The inference system is illustrated as "inference system 130" in Figs. 1 and 4:

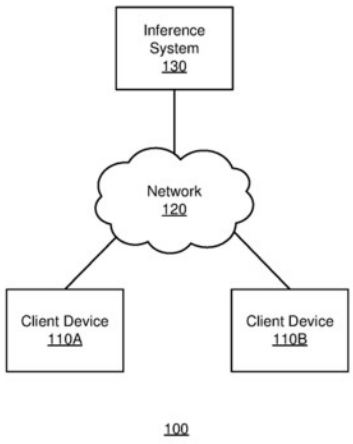


FIG. 1

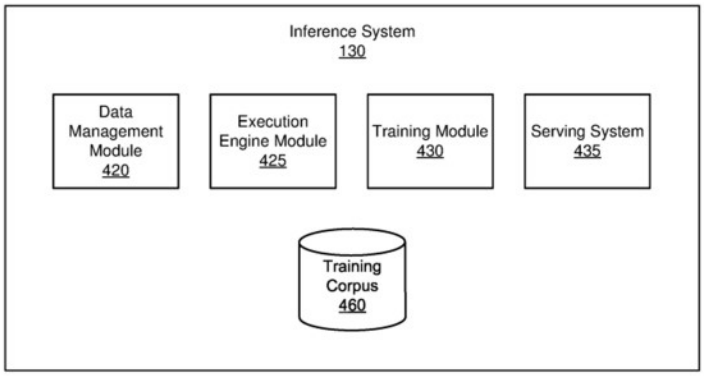


FIG. 4

Ex. 2, Figs. 1 & 4.

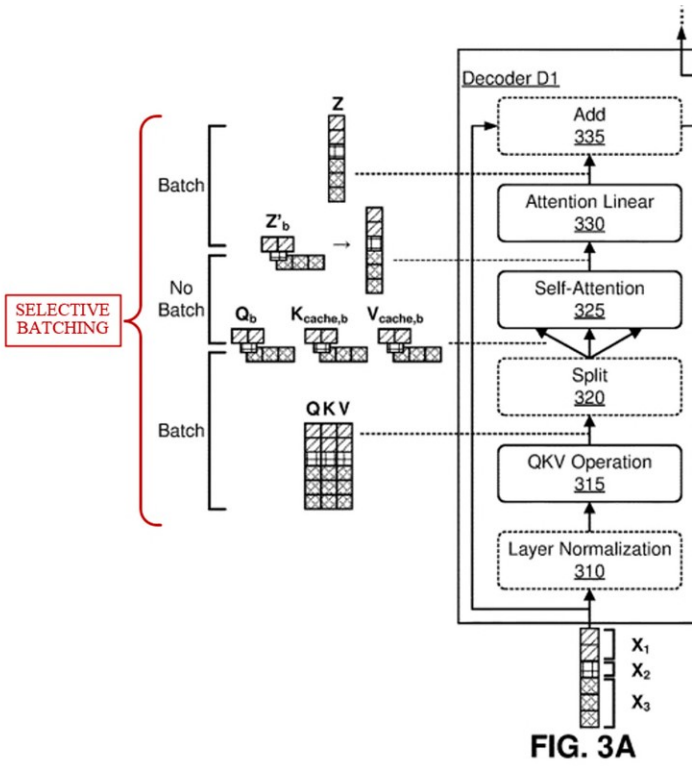
[1] Unless otherwise noted, citations to the asserted patents refer to the parent '775 patent (Ex. 2).

Referring to Fig. 4, the inference system includes physical components such as an "Execution Engine Module 425" and a "Serving System 435" which include "central processing unit (CPU) cores, CPU memory (*e.g.*, DRAM), data storage, [and] one or more execution engines (*e.g.*, GPU devices)." *Id*. at 5:7-12; *see also id*. at 19:41-44, 5:12-22, 20:4-7. Figs. 5A-5D illustrate the components for "Serving System 435," which include a "Request Processor 580" and a "Scheduler 585" that are each "coupled to" "one or more execution engines." *Id*. at Figs. 5A-5D, 22:23-46.

The asserted patents also describe two batching processes for batching input requests with variable lengths (*e.g.*, input sentences having a different number of words). *See id*. at Abstract, 1:46-51, 1:55-61, 2:50-57, 10:34-37. The first batching process is called "selective batching" as shown in Figs. 3A-3B. *Id*. at Figs. 3A-3B, 10:33-22:21. The second batching process is called "Dynamic Batching" as shown in Figs. 5A-5D." *Id*. at Figs. 5A-5D, 22:22-24:38; *see also* Ex. 7 ('775 patent prosecution history) at 189 (May 3, 2022 Amendment). Importantly, during prosecution FriendliAI expressly argued the claimed invention for both patents requires *both* batching processes. *Id*.; *see also* Ex. 8 ('520 patent file history) at 284 (Aug. 7, 2023 Amendment).

1. Selective Batching

The selective batching process involves a transformer model (executed by the execution engine) that selectively batches or groups multiple inputs for "a subset of operations in the transformer model." Ex. 2 at 11:12-17. Importantly, the asserted patents teach that some transformer model operations are not compatible with batching and require the inputs to be processed individually (*i.e.*, not batched). In other words, the transformer model itself performs selective batching by batching input requests for specific subsets of operations:



Ex. 2, Fig. 3A
(annotated)

Fig. 3A illustrates the transformer model selectively batching multiple input sequences X_1 ("what's up"), X_2 ("hello") and X_3 ("how are you") and executing specific operations on the input tokens/words either together ("Batch") or individually ("No Batch"). *Id*. at 11:31-40, 11:12-17, 11:41-47, 11:41-47. To effectuate this, the batch operation performed by the transformer model concatenates, or combines, multiple input sequences (X_1, X_2, X_3) together for specific operations. *Id*. at 11:19-24 ("By selective batching, the inference system 130 can allow batching operations to be performed for a batch of requests with variable input or target length or internal state length to utilize the parallel computation capabilities of hardware accelerators."), 11:54-56 ("[I]n the selective batching method the inference system 130 may concatenate the input token sequences X_1, X_2, and X_3 into a concatenated input tensor."), 11:56-60 ("Different from the batching method in FIGS. 2A-2B, the inference system 130 concatenates the input

token sequences such that individual input tokens for the batch of requests are concatenated across one dimension.").

2. Dynamic Batching

The dynamic batching process relates to a scheduler that "adjust[s] . . . batches at an iteration." *Id*. at 24:10-13. An iteration refers to a single pass of the transformer model, where the model processes a batch of input data and computes an output. *Id*. at 2:9-13; *see also id*. at Fig. 3A (one "iteration" through transformer model 300 begins with inputs X_1, X_2, X_3 and ends with outputs $\hat{y}_1$). During dynamic batching, the scheduler monitors the cache memory available on the execution engines and allocates new request batches to an execution engine with sufficient memory available. *Id*. at 23:22-29; *see also id*. at 22:40-46. Put simply, the dynamic batching process is a "load-balancing" function, as shown in Figs. 5A-5D. *Id*. at Figs. 5A-5D, 22:22-24:38.

Depicted in Figs. 5A-5D, scheduler 585 monitors physical memory (KV cache) and the batches being executed by various execution engines 590A-B. *Id*. at Figs. 5A-5D. At the end of an iteration, the execution engine's available memory can increase or "free up" if the execution engine provides a final output response for a corresponding input request in a batch and removes that input request from its local batch. *Id*. at 23:51-67 ("[T]he first output token generated for request R2 is generated with an end token and the execution engine 590A provides the outputs for request R2 to the completion queue of the request processor 580. The execution engine 590A frees the cache memory allocated to request R2.").

Figs. 5C-5D further highlight the dynamic batching process:

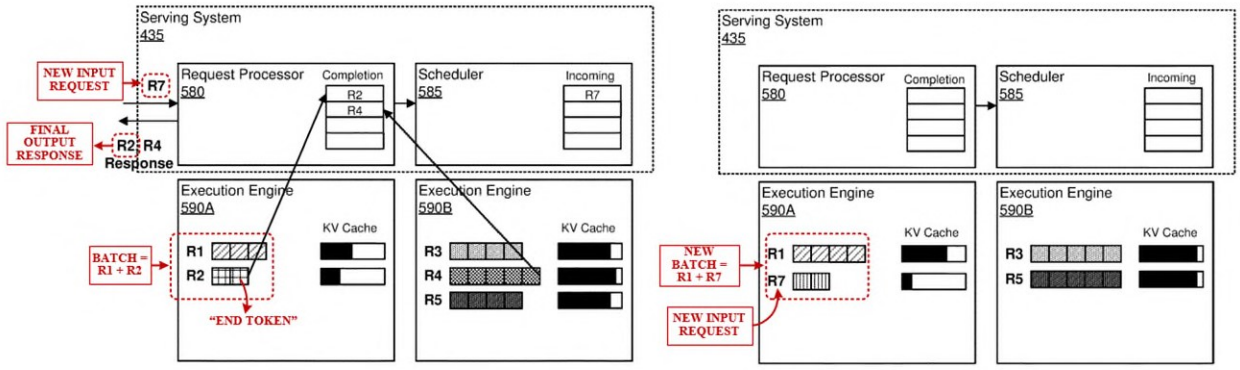


FIG. 5C FIG. 5D

Ex. 2, Figs. 5C & 5D

(annotated)

In Fig. 5C, execution engine 590A processes a batch that includes the input requests R1 and R2. *Id*. at Fig. 5C. At the end of an iteration, the transformer model (executed by execution engine 590A) outputs a final "end token" for request R2. *Id*. Execution engine 590A then provides the outputs for request R2 to the "completion" queue in request processor 580, removes request R2 from its batch, and frees up the physical memory (KV cache) allocated to request R2. *Id*. Notably, scheduler 585 also receives and stores a new input request R7 in its "incoming" request queue. *Id*. at Fig. 5C, 23:51-24:16. In the subsequent Fig. 5D, scheduler 585 then determines how much memory in execution engine 590A is available and schedules a new batch that includes requests R1 + R7. *Id*. at 24:17-27. In this fashion, the dynamic batching process schedules and load balances the batches being executed by respective execution engines based on available memory.

B. Prosecution History

During prosecution, the Examiner rejected the claims for both asserted patents and issued a one respective Office Action. *See* Ex. 7 at 117-125 (Feb. 9, 2022 Non-final Office Action); Ex. 8 at 116-132 (Feb. 14, 2023 Non-final Office Action). FriendliAI responded to the Office

Actions with similar amendments and arguments.

Specifically, FriendliAI amended the claims of the '775 patent to further recite "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. 7 at 181-186 (May 3, 2022 Amendment). In the corresponding remarks, FriendliAI explained the claimed invention requires (1) a serving system, (2) a scheduler, (3) an execution engine, and (4) a request processor. *Id*. at 188-189. Importantly, FriendliAI expressly argued the claimed invention distinguishes over the cited prior art as follows:

> [B]y performing the *dynamic batching method* described in conjunction with FIGs. 5A-5D of the application with a transformer architecture *with selective batching* (e.g., described in conjunction with FIGs. 3A-3B), an inference system can modify a batch being processed and update the batch between iterations as new requests are received, even if a length of the input for the new request is different from a length of the input for another request in the batch.

Id. at 189. Thus, according to FriendliAI, the invention requires and incorporates both the dynamic batching process performed by the scheduler and described in Figs. 5A-5D and the selective batching process performed by the transformer model and described in Figs. 3A-3B.

During prosecution of the child '520 patent, FriendliAI filed a preliminary amendment incorporating substantially similar claim recitals as those provided in connection with the '775 patent. *See* Ex. 8 at 94-100 (Sept. 9, 2022 Preliminary Amendment). The Examiner issued a double patenting rejection over the claims of the '775 patent, and FriendliAI responded by filing a terminal disclaimer. *Id*. at 118-124 (Feb. 14, 2023 Office Action), 254-255 (Aug. 7, 2023 Terminal Disclaimer). FriendliAI further responded to the Examiner's prior art rejections by again arguing the claimed invention requires and incorporates both the dynamic batching process described in Figs. 5A-5D *and* the selective batching process described in Figs. 3A-3B. *Id*. at 284 (Aug. 7, 2023 Amendment) ("However, by performing the dynamic batching method described

in conjunction with FIGs. 5A-5D . . . with selective batching (e.g., described in conjunction with FIGs. 3A-3B."). The Examiner relied on FriendliAI's amendments and representations regarding the two batching processes and issued the respective patents. *See* Ex. 7 at 195-96 (Notice of Allowance); Ex. 8 at 292-93 (Notice of Allowance.)

C. The Claimed Invention

The asserted claims for both asserted patents are directed to the same invention and require specific, distinct hardware components, including a "serving system," an "execution engine," and some kind of "scheduler" to perform the selective and dynamic batching processes. The relevant recitals for representative claim 1 for each asserted patent are below:

[**selective batching by the transformer model in the execution engine**]

> 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*"

Ex. 2, claim 1 (emphasis added); Ex. 1, claim 1 (emphasis added).

[**dynamic batching by the scheduler**]

> *scheduling, by the scheduler, a second batch of requests . . . for execution on the execution engine . . . 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. 2, claim 1 (emphasis added); Ex. 1, claim 1 (emphasis added).

Plaintiff's Reply Asserted Patents

As an initial matter, Defendant makes several incorrect statements in the Background of the Asserted Patents section, while misleadingly citing to the patent specifications. FriendliAI takes the opportunity to address and clarify such statements here.

First, Defendant refers to various components and systems as "physical," citing to figures and excerpts from the specification. (*See* Defendant's Responsive Claim Construction Brief

("Resp.") at 1-2 ("the inference system includes physical components. . ." (citing Ex. 2 at 5:7-12, 19:41-44, 5:12-22, 20:4-7)), 4 ("scheduler 585 monitors physical memory . . ." (citing Ex. 2 (U.S. Patent No. 11,442,775 ("'775 Patent") at Figs. 5A-5D)), 5 ("Execution engine 590A . . . frees up the physical memory . . ." (citing Ex. 2 at 23:51-24:16)); *see also* Resp. at 7 (stating that claimed inventions "require specific, distinct ***hardware components*** . . .")(emphasis added)). But none of Defendant's cited figures or excerpts from the specification refers to the component or system as being "physical," let alone requiring "specific, distinct hardware components."

Similarly, Defendant describes the dynamic batching process of the asserted patents as "a 'load-balancing' function," again citing to the patent specifications. (Resp. at 4 (citing Ex. 2 at Figs. 5A-5D, 22:22-24:38.) Again, however, the specification never refers to, or otherwise supports the notion that, dynamic batching is merely "load balancing."

Defendant also inaccurately states that "***the batch operation*** . . . concatenates, or combines, multiple input sequences (X1, X2, X3) together," again purporting to cite to the patent specifications. (Resp. at 3 (emphasis added) (citing Ex. 2 at 11:19-24, 11:54-56, 11:56-60).) Again, none of Defendant's cited specification excerpts describes combining or concatenating input sequences as a "batch operation." As discussed in FriendliAI's opening brief and in further detail below, a "batch operation" is not a process for combining or concatenating multiple input sequences, but rather refers to the execution of a transformer model operation ***on a batch*** of requests.

FriendliAI respectfully submits that Defendant's unsupported and incorrect statements in its background section should be disregarded.[2]

[2] Defendant also makes a number of incorrect statements regarding selective batching, including purporting to identify the "recitals of representative claim 1" that correspond to selective batching (Resp. at 7), and otherwise describing the selective process in a manner that is unsupported by the

III. LEVEL OF ORDINARY SKILL IN THE ART

Plaintiff's Opening

A person of ordinary skill in the art ("POSITA") of the invention claimed in the '775 and '520 Patents is a software engineer with a master's in computer science or related discipline, and at least two years of experience.

Defendant's Answering

A person of ordinary skill in the art (POSITA) would have either (1) a bachelor's degree in electrical engineering, computer engineering or computer science, with two to three years of work experience in machine learning, or (2) a master's degree in electrical engineering, computer engineering or computer science, with one year of work experience in machine learning. Other combinations of education with relevant practical experience may also qualify one as a POSITA, as the case may be.

IV. LEGAL FRAMEWORK

Plaintiff's Opening

A. Claim Terms are Presumed to Have Their Plain and Ordinary Meanings

Claim construction is a matter of law exclusively for the Court. *Teva Pharms. USA, Inc. v. Sandoz, Inc.,* 574 U.S. 318, 325 (2015). Courts interpret claims and their terms as they would have been understood by a person of ordinary skill in the art at the time of the invention. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313 (Fed. Cir. 2005) (en banc). "Importantly, the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the

specification. (Resp. at 2-3.) While these statements do not pertain specifically to the claim construction issues and thus FriendliAI does not address them in detail here, FriendliAI nonetheless notes that Defendant's descriptions are unsupported and inaccurate.

specification." *Id*. "The specification is, thus, the primary basis for construing the claims." *Id.* at 1315. In addition to the specification, the Court should consider the prosecution history of the patents, which "consists of the complete record of the proceedings before the PTO and includes the prior art cited during the examination of the patent." *Id.* at 1317. Extrinsic evidence, such as dictionary definitions, may be helpful to come to a proper understanding of the term, but usually "an analysis of the intrinsic evidence alone will resolve any ambiguity in a disputed claim term." *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1583 (Fed. Cir. 1996).

Claim terms "are generally given their ordinary and customary meaning," which is the "meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention." *Phillips*, 415 F.3d at 1312–13 (citations omitted). Courts must not disturb the plain and ordinary meaning of claim terms unless required by (1) an express definition in the specification, or (2) a clear and unmistakable disavowal of the full scope of the claim terms in the intrinsic record. *Hill-Rom Servs., Inc. v. Stryker Corp.*, 755 F.3d 1367, 1371 (Fed. Cir. 2014) ("We depart from the plain and ordinary meaning of claim terms based on the specification in only two instances: lexicography and disavowal."); *Merck & Co. v. Teva Pharm. USA, Inc.*, 395 F.3d 1364, 1370 (Fed. Cir. 2005).

B. Indefiniteness Must be Proven by Clear and Convincing Evidence

A patent is not invalid for indefiniteness unless "its claims, read in light of the specification delineating the patent, and the prosecution history, fail to inform, with reasonable certainty, those skilled in the art about the scope of the invention." *Nautilus, Inc. v. Biosig Instruments, Inc.*, 572 U.S. 898, 901 (2014). Indefiniteness, like all other invalidity defenses, must be proven by clear and convincing evidence. *Biosig Instruments, Inc. v. Nautilus, Inc.*, 783 F.3d 1374, 1377 (Fed. Cir. 2015). "The definiteness requirement must take into account the inherent limitations of language. 'Some modicum of uncertainty . . . is the price of ensuring the appropriate incentives for

innovation.'" *Teva Pharm. USA, Inc. v. Sandoz, Inc.*, 789 F.3d 1335, 1341 (Fed. Cir. 2015) (quoting *Nautilus*, 572 U.S. at 910).

Defendant's Answering

A. Claim Construction

"It is a bedrock principle of patent law that the claims of a patent define the invention to which the patentee is entitled the right to exclude." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (internal quotation and citation omitted). "'[T]here is no magic formula or catechism for conducting claim construction.' Instead, the court is free to attach the appropriate weight to appropriate sources 'in light of the statutes and policies that inform patent law.'" *SoftView LLC v. Apple Inc.*, 2013 WL 4758195, at *1 (D. Del. Sept. 4, 2013) (quoting *Phillips*, 415 F.3d at 1324). When construing patent claims, a court considers the literal language of the claim, the patent specification, and the prosecution history. *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 977-80 (Fed. Cir. 1995), aff'd, 517 U.S. 370 (1996). "[T]he specification is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term." *Phillips*, 415 F.3d at 1315 (internal quotation and citation omitted).

When a court relies solely upon the intrinsic evidence—the claims, the specification, and the prosecution history—the court's construction is a determination of law. *See Teva Pharmaceuticals USA, Inc. v. Sandoz, Inc.*, 574 U.S. 318, 331 (2015). "A claim construction is persuasive, not because it follows a certain rule, but because it defines terms in the context of the whole patent." *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998). It follows that "a claim interpretation that would exclude the inventor's device is rarely the correct interpretation." *Modine Mfg. Co. v. U.S. Int'l Trade Comm'n*, 75 F.3d 1545, 1550 (Fed. Cir. 1996).

Courts may also make findings based upon extrinsic evidence, which "consists of all evidence external to the patent and prosecution history, including . . . dictionaries." *Phillips*, 415 F.3d at 1317-19 (internal quotation and citation omitted). Extrinsic evidence may assist the court in understanding the underlying technology, the meaning of terms to one skilled in the art, and how the invention works. *Id*. Extrinsic evidence, however, is less reliable and less useful in claim construction than the patent and its prosecution history. *Id*.

B. Indefiniteness

"[A] patent is invalid for indefiniteness if its claims, read in light of the specification delineating the patent, and the prosecution history, fail to inform, with reasonable certainty, those skilled in the art about the scope of the invention." *Nautilus, Inc. v. Biosig Instruments, Inc.*, 572 U.S. 898, 901 (2014) (2014). A patent must be precise enough to afford clear notice of what is claimed. *Id*. at 2129. While "clear and convincing evidence" is required to show that a term is indefinite, "this presumption of validity does not alter the degree of clarity that § 112, ¶ 2 demands from patent applicants." *Id*. at 912 n.10.

V. AGREED-UPON CLAIM TERMS

The parties agree on proposed constructions for the following claim terms:

Claim Term	Joint Proposed Construction
"a second batch of requests"	"a batch that includes at least one request from the [first] batch of requests"
"length" terms (*See* D.I. 109 ("Revised Joint Claim Construction Chart").)	"length" refers to a "number or quantity of <units>" for a given context—e.g., the number of "tokens" in a sequence, such as "a [number or quantity of tokens] of the sequence of input tokens for the new request is different from a [number or quantity of tokens] of an input for at least one request"
"a new request"	"a request other than the one or more requests of the [first] batch of requests"

VI. DISPUTED CLAIM TERMS

A. "dedicated cache memory"/ "cache memory dedicated"

Claim Term	FriendliAI's Proposed Construction	Defendant's Proposed Construction
"dedicated cache memory"/ "cache memory…dedicated"	Plain and Ordinary meaning. In the alternative: "memory for storing an internal state for a request"	"physical memory in, or a component part of, the execution engine"

1. Plaintiff's Opening Position

Dependent claims 3, 5, 12, and 14 of the '775 Patent, and dependent claims 4, 6, 14, 16, 23, 24, 30, 31, 37, 39, 45, 47, 53, 55, 61, and 63 of the '520 Patent recite the terms "dedicated cache memory" and/or "cache memory … dedicated." Because this term is clear from the surrounding claim language and the specification, FriendliAI respectfully submits that construction of this term is unnecessary. In the alternative, to the extent that the Court determines that the claim term requires construction, FriendliAI proposes the construction of "memory for storing an internal state for a request," which comports with the plain language of the asserted claims and the intrinsic record.

Claim language does not require construction where the language carries its ordinary meaning, and that meaning is clear from the claim. *Chef Am., Inc. v. Lamb–Weston, Inc.*, 358 F.3d 1371, 1375 (Fed. Cir. 2004) ("[W]e have repeatedly declined to rewrite unambiguous patent claim language"); *Mentor H/S, Inc. v. Med. Device Alliance, Inc.*, 244 F.3d 1365, 1380 (Fed. Cir. 2001) (there is a "heavy presumption" in favor of ordinary meaning). Here, the meaning of "dedicated cache memory" and "cache memory…dedicated" is clear from the claim language and specification, and therefore these terms need no construction. Specifically, the claims identify precisely what the cache memory is "dedicated" to storing—that is, either "an internal state" or

the "key cache tensor" and "value cache tensor." For example, claims 3 and 5 of the '775 Patent are provided below:

> Claim 3: The method of claim 2, wherein the request is associated with a ***cache memory in the execution engine dedicated*** for storing an internal state for the request, and responsive to determining that the request has been completed, freeing the ***dedicated cache memory*** for the request in the execution engine.
>
> Claim 5: The method of claim 4,
>
> wherein the execution engine includes a ***cache memory*** for maintaining a key cache tensor for storing keys and a value cache tensor for storing values for the at least one request, and
>
> wherein after scheduling the second batch of requests, allocating, by the execution engine, a new ***cache memory dedicated*** to maintaining a key cache tensor and a value cache tensor for the new request.

As shown above claims 3 of the '775 Patent recites on its face that that the "cache memory" is "dedicated" to "storing an internal state." (Ex. 2, cl. 3; *see also* Ex. 2, cl. 12; Ex. 1, cls. 4, 14, 23, 30, 37, 45, 53, and 61). Similarly, as shown above, claim 5 of the '775 patent recites that the "cache memory" is "dedicated" to "maintaining" a "key cache tensor" and a "value cache tensor." (Ex. 2, cl. 5; *see also* Ex. 2, cl. 14; Ex. 1, cl. 6, 16, 24, 31, 39, 47, 55, and 63.)

Moreover, the plain meaning of "dedicated cache memory" and "cache memory . . . dedicated" is readily apparent from the specification. For example, the specification explains that in transformer models, the internal state may be represented by the "key cache tensor" and the "value cache tensor." (*See, e.g.*, Ex. 1 at 23:4-15 (describing that the "***key cache tensor and a value cache tensor*** are allocated for each request ***as the internal state cache***, and ***part of the cache for each request is used to store the keys and values*** after the encoding phase." (emphasis added).) The "key cache tensor" and the "value cache tensor" are commonly referred to as the KV ***Cache***, as depicted in Figs. 5A-5D. (*See* Ex. 1, Figs. 2A-2B, 3A-3B, 5A-5D.) Accordingly, a POSITA would readily understand in light of the specification that the use of the word "cache" in "cache

memory" refers to the KV cache—also known as the "key cache tensor" and a "value cache tensor," or the "internal state." This is consistent with the plain language of the claims, which, again, expressly describes the "cache memory" as dedicated to storing the "internal state cache," or the "key cache tensor" and "value cache tensor."

In contrast, Defendant's proposed construction inserts extraneous, unsupported limitations in an attempt to improperly redefine the plain meaning of the claim terms. As the Federal Circuit has repeatedly instructed, adding limitations that depart from the plain and ordinary meaning of the claim terms is improper. *Johnson Worldwide Assocs., Inc. v. Zebco Corp.*, 175 F.3d 985, 989 nn.3-4 (Fed. Cir. 1999) (in construing claims, "[g]eneral descriptive terms will ordinarily be given their full meaning; modifiers will not be added to broad terms standing alone."); *GE Lighting Sols., LLC v. AgiLight, Inc.*, 750 F.3d 1304, 1309 nn.7-8 (Fed. Cir. 2014) ("[I]t is improper to read limitations from a preferred embodiment described in the specification—even if it is the only embodiment—into the claims absent a clear indication in the intrinsic record that the patentee intended the claims to be so limited." (quoting *Liebel–Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 913 (Fed.Cir.2004))). Here, Defendant proposes construing "dedicated cache memory" as "***physical*** memory ***in, or that is a component part of***, the execution engine." (D.I. 109 (Revised Joint Claim Construction Chart).) In other words, Defendant proposes to add two improper limitations: Defendant seeks to limit the claimed memory to a "physical" memory, and further seeks to limit the claimed memory to one that is "in, or a component part of, the execution engine."

Both limitations are inconsistent with the plain meaning of the terms and unsupported by the intrinsic record, and are therefore improper. First, the addition of the term "physical" is inconsistent with the plain meaning of the term "memory." (Ex. 3 at ¶ 38; Ex. 4 (Newton's Telecom Dictionary (30th ed. 2016)) at 806 ("*memory*: The part of a computer or sophisticated

phone system which stores information or instructions for use. Memory comes in many variations. There is memory which is lost when the power is switched off. There is memory which is retained when power is turned off.").) The specification never refers to a "physical memory." (*See generally* Ex. 1 and Ex. 2.) To the contrary, the specification describes multiple types of memory structures that can exist as part of an execution engine, none of which are limited to "physical" memory. (*See, e.g.*, Ex. 1 at 19:48-20:31.) Additionally, the specification describes scenarios where an "execution engine may be composed of one or more hardware accelerators" and the "execution engine is able to distribute the workload across multiple hardware accelerators if necessary." (Ex. 1 at 20: 4-10.) In other words, the specification contemplates that the claimed system utilizes a distributed system, which necessarily incorporates non-physical components. (Ex. 3 at ¶ 38.)

Further, Defendant's recitation of "in, or a component part of" is superfluous, ambiguous and unsupported by the specification. As an initial matter, the plain language of the claims already specifies that the claimed cache memory is "in the execution engine." (*See, e.g.*, Ex. 2, cl. 3 ("The method of claim 2, wherein the request is associated with a cache memory ***in the execution engine*** dedicated for storing an internal state for the request . . ." (emphasis added)); *see also* Ex. 1, cl. 4 ("cache memory ***in the execution engine*** dedicated" (emphasis added)); Ex. 1, cl. 6 ("wherein the execution engine includes a cache memory.").) In sum, there is no need to define the cache memory in terms of the execution engine since the claim language itself recites the relationship.

Defendant's proposal that the claimed cache memory can be "a component part of" an execution engine is not only unsupported by the intrinsic record, but also introduces confusion. Indeed, it is unclear how memory, particularly Defendant's proposed "physical memory," can be a "component part of" the execution engine without being in the execution engine. "Importantly,

a claim construction, if needed at all, should help resolve, not add to, uncertainty in the understanding the finder of fact is to use in applying a claim term." *Promptu Sys. Corp. v. Comcast Corp.*, 92 F.4th 1372, 1381 (Fed. Cir. 2024). The specification does not disclose the component parts of the execution engine, so it is unclear what Defendant expects to include within that construction.

Finally, Defendant's proposed construction improperly attempts to read out the words, "dedicated" and "cache," from the disputed claim term. *Bicon, Inc. v. Straumann Co.*, 441 F.3d 945, 950 (Fed. Cir. 2006) (explaining that to read limitations out of a claim would "be contrary to the principle that claim language should not be treated as meaningless"). Specifically, Defendant proposes to construe "dedicated cache memory" and "cache memory . . . dedicated." But notably, nothing in Defendant's proposed construction—*i.e.*, "physical memory in, or a component part of, the execution engine"—addresses the words "dedicated" or "cache." Instead, Defendant's construction merely attempts to limit "memory" to being "physical" and as being "in, or a component part, the execution engine." Such a construction imports unsupported limitations into the claims and disregards portions of the claim language, and is therefore improper.

FriendliAI respectfully submits that construction of this term is unnecessary. The claims and specification are clear as to the meaning of the "dedicated cache memory." Should the Court believe that construction is necessary, FriendliAI submits that the term should be construed consistent with its plain meaning: as "memory for storing an internal state for a request."

2. Defendant's Answering Position

The heart of this dispute relates to whether the recited memory requires a physical structure in the execution engine. Hugging Face asserts the "cache memory" refers to a physical memory in the execution engine, while FriendliAI proposes the memory could include "non-physical components" and further defines the term according to additional functional language,

namely "for storing an internal state for a request." *See supra* at 17, 19.

The dedicated "cache memory" term appears in two sets of claims:

[**Set 1**]:

"a *cache memory* in the execution engine *dedicated* for storing an internal state for the request"

Ex. 2, claims 3, 12 (emphasis added); Ex. 1, claims 4, 14, 23, 27, 30, 45, 53, 61 (emphasis added).

[**Set 2**]:

"the execution engine includes a *cache memory* for maintaining a key cache tensor" and "allocating, by the execution engine, a new *cache memory dedicated* to maintaining a key cache tensor"

Ex. 2, claims 5, 14 (emphasis added); Ex. 1, claims 6, 16, 24, 31, 39, 47, 55, 63 (emphasis added).

The relevant claim language recites "cache memory" "*in* the execution engine" (Set 1) and the "execution engine *includes* a cache memory" (Set 2). In accordance, the parties appear to agree the plain claim language requires the dedicated "cache memory" to be part of, or reside in, the execution engine. *See supra* at 20 ("[T]he claim language itself recites the relationship.").

Turning to the dispute over the physical memory requirement, FriendliAI argues the "cache memory" hardware could somehow include unknown and undefined "non-physical components." *Id*. But the intrinsic evidence, including the plain claim language and the specification, consistently describe the "cache memory" as a physical memory structure in the execution engine.

First, the plain claim language, as informed by the specification, refers to physical

memory. The claims refer to the physical position of the "cache memory" relative to the "execution engine," where the memory must reside "in" the execution engine or be "include[d]" as part of the execution engine. If the "cache memory" was non-physical, it would not reside "in" the execution engine, nor would it make sense for the execution engine to "include" non-physical memory. This is so because the execution engine itself, as defined by the asserted patents, is a physical hardware component (*e.g.*, "GPU devices," "graphics processing units (GPU's)," or "tensor processing units (TPU's)"). *Id*. at 19:41-44 ("The execution engine module 425 includes [sic] or more execution engines that are built on specialized hardware accelerators such as graphics processing units (GPU's) or tensor processing units (TPU's)."); *see also id*. at 1:32-37 ("the inference system executes the requests on specialized hardware accelerators such as graphics processing units (GPU's) or tensor processing units (TPU's) to improve latency and throughput"), 4:60-63 ("the inference system 130 includes one or more execution engines that are built on specialized hardware accelerators such as graphics processing units (GPU's) or tensor processing units (TPU's)"), 5:7-12, 14:65-15:8. Therefore, to reside "in" or be "include[d]" in a physical execution engine, the cache memory must itself be physical hardware. *Id*. at 19:49-51 ("As described above, an execution engine may include a set of cores (e.g., GPU 50 cores) coupled to local memory (e.g., GPU memory).").

This is consistent with how FriendliAI's own expert describes the cache memory, as he states "[t]he claims explain that the cache memory *in* the execution engine is dedicated for storing either the internal state or the key cache tensor and the value cache tensor." Ex. 3 at ¶ 36 (emphasis added). Similarly, FriendliAI's expert opines that "[t]he claims also describe the cache memory as being 'in the execution engine.'" Ex. 3 at ¶ 37 (quoting the claim language). Further, in opining the execution engines "would be understood by a POSITA to include non-

physical components," he cites solely to the specification's disclosure of *hardware* (*i.e.*, physical) aspects of the execution engines. Ex. 3 at ¶ 38.

The specification also describes operations performed by the execution engine to store data "on" its cache memory—"during training or inference of the transformer model, data required for inference or training is read from an input file in data storage by the CPU or across network 120 . . ., moved to local memory of an execution engine, and processed by the execution engine." *Id*. at 5:22-27. Further, the "execution engine is . . . configured to manage one or more caches on the local memory." *Id*. at 20:4-5. Finally, Fig. 8 illustrates various hardware components, such as "Main Memory 803," which represents "a random-access memory (RAM) or other dynamic storage device, for storing information and instructions to be executed by the processor 801." *Id*. at Fig. 8, 25:56-60. The description of the physical, local "main memory 803" hardware is consistent with a physical, local "cache memory." *Id*.; *see also id*. at 20:4-7.

Second, the extrinsic evidence also supports Hugging Face's proposed construction. FriendliAI's definition of generic "memory" from Newton's Telecom Dictionary confirms "memory" is a physical structure as it is "part of a computer." *See supra* at 19-20. Additional dictionary definitions consistently define "cache memory" as a physical structure. (Ex. 9, *The American Heritage Dictionary of the English Language*, 4th ed. ("cache . . . 2. *Computer Science* A fast storage buffer in the central processing unit of a computer. Also called *cache memory*."); Ex. 10, *Encyclopedia Britannica*, May 20, 2021 (cache memory – "[b]oth main memory and cache are internal random-access memories (RAMs) that use semiconductor-based transistor circuits"); Ex. 11, *The Authoritative Dictionary of IEEE Standard Terms*, 7th ed. (cache memory – "[a] buffer memory inserted between one or more processors and the bus, which is used to hold currently active copies of blocks of information from main memory"); Ex. 12, *A Dictionary of*

Computer Science, 7th ed. (cache memory – "[a] type of memory that is inserted between the processor and memory proper.").

Turning to FriendliAI's alternate functional definition, it copies and parrots the surrounding claim language in only the Set 1 claims and ignores the differing claim language in the Set 2 claims. *Compare* Ex. 2, claim 3 (Set 1), *with* Ex. 2, claim 5 (Set 2). Repeating the claim language in this fashion is superfluous and does not align with the language of the Set 2 claims. FriendliAI implicitly acknowledges this contradiction in arguing the "cache memory" can *either* (1) store an internal state (Set 1) *or* (2) store the key cache tensor and the value cache tensor (Set 2). *See supra* at 17-18.

FriendliAI's remaining argument recasts the dispute by focusing on the adjective "dedicated." *Id*. at 10. But this argument misses the bigger picture, as "dedicated" does not change the fact that the "cache memory" requires a physical structure in, or part of, the execution engine.

The plain claim language, the written description, and extrinsic dictionary definitions of "cache memory" confirm Hugging Face's construction of the disputed "cache memory" terms. The Court should thus adopt Hugging Face's construction in favor of FriendliAI's alternative proposals.

3. Plaintiff's Reply Position

Defendant's proposed construction for "dedicated cache memory" and "cache memory dedicated" attempts to add an unsupported, extraneous limitation requiring the claimed memory to be "physical," while reading out the claim terms "dedicated" and "cache." Such a construction violates fundamental canons of claim construction, and should be rejected.

Defendants admit that the "heart of this dispute relates to whether the recited memory requires ***a physical structure*** in the execution engine." (*Supra* p. 21 (emphasis added).) But such

a requirement for "physical structure" is also inconsistent with the plain meaning of the term "memory." Indeed, as explained in FriendliAI's opening brief and Dr. Pazzani's expert declaration, memory comes in a variety of forms—many of which are not "physical," such as virtual memory or cloud-based storage. (*See* Ex. 3 (Expert Declaration of Michael Pazzani) at 12 ("there exist memory that is non-physical (e.g., cloud-based storage, virtual memory)"); (*Supra* p. 17.)

Further, Defendant's proposed "physical" limitation is inconsistent with the specification of the asserted patents. The term "physical" appears nowhere in the specification or claims, and, as explained in FriendliAI's opening brief, the specification even contemplates a system with non-physical components. (*Supra* pp. 19-20.) Defendant also acknowledges in its response that the specification of the asserted patent refers to "a random-access memory (RAM) ***or other dynamic storage device***," which in turn is described as "a magnetic disk or optical disk or solid-state memory device." (*Supra* p. 24 (emphasis added); Ex. 2 at 25:56-60, 26:3-5.)[3] Such a dynamic storage device, such as a solid-state memory device (SSD), may include non-physical, cloud-based or virtual memory.[4]

[3] To the extent that Hugging Face argues that the term "memory" should be limited to a "random-access memory (RAM)," at the exclusion of other dynamic storage devices that reflect non-physical memory, the Court should reject such arguments that attempt to improperly "read[] a limitation from the written description into the claims." *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1320 (Fed. Cir. 2005); *see also GE Lighting Sols., LLC v. AgiLight, Inc.*, 750 F.3d 1304, 1309 (Fed. Cir. 2014) ("[I]t is improper to read limitations from a preferred embodiment described in the specification—even if it is the only embodiment—into the claims absent a clear indication in the intrinsic record that the patentee intended the claims to be so limited." (quoting *Liebel–Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 913 (Fed. Cir. 2004))).

[4] *See, e.g.*, https://www.lenovo.com/us/en/glossary/physical-memory/ (comparing "virtual memory" and "physical memory," and explaining that "[p]hysical memory (RAM) is the actual hardware that provides temporary storage for active data, while ***virtual memory is a reserved space on the storage device (usually a hard drive or SSD [solid-state drive]***)" (emphasis added)).

As the Federal Circuit has repeatedly instructed, importing claim limitations that narrow a claim term beyond its plain and ordinary meaning is improper, and this Court should thus reject Defendant's attempt to limit the claim term "memory" to "physical" memory. *See 3M Innovative Properties Co. v. Tredegar Corp.*, 725 F.3d 1315, 1333 (Fed. Cir. 2013) ("It is axiomatic that we will not narrow a claim term beyond its plain and ordinary meaning unless there is support for the limitation in the words of the claim, the specification, or the prosecution history." (citing *Douglas Dynamics, LLC v. Buyers Prods. Co*., 717 F.3d 1336, 1342 (Fed. Cir. 2013); *Aventis Pharma S.A. v. Hospira, Inc.*, 675 F.3d 1324, 1329 (Fed. Cir. 2012))); *see also Johnson Worldwide Assocs., Inc. v. Zebco Corp.*, 175 F.3d 985, 989 (Fed. Cir. 1999) (in construing claims, "[g]eneral descriptive terms will ordinarily be given their full meaning; modifiers will not be added to broad terms standing alone.").

Moreover, Defendant's argument that the claimed memory must be physical because it is "***in*** the execution engine," is unavailing. (*Supra* pp. 21-24 (emphasis added).) Again, non-physical memory, such as virtual memory, can be "in" an execution engine without being physical memory, as there are different types of non-physical memory, such as virtual memory, that may be in an execution engine. (*See* Ex. 3 at 12; *supra* p. 20.) Moreover, Defendant's argument that the "execution engine itself, as defined by the asserted patents, is a physical hardware component", is incorrect. (*Supra* p. 23.) Dr. Pazzani explained that the specification contemplates an execution engine that is "composed of one or more hardware accelerators" and "is able to distribute the workload across multiple hardware accelerators if necessary." (*See* Ex. 3 at 12 (citing Ex. 1 (U.S. Patent No. 11,836,520 ("'520 Patent")) at 20:4-10).) Such a system "would be understood by a POSITA to include non-physical components." (Ex. 3 at 12.) In other words, where an execution engine is associated with multiple hardware accelerators (as the specification contemplates), there

may be memory associated with each hardware accelerator which is mapped to virtual, cloud-based or other non-physical memory in the execution engine. In sum, Defendant's unsupported arguments—both that the execution engine is necessarily physical and that the presence of memory "in" the execution engine requires the memory to be physical should be rejected. (*Supra* p. 23.)

The Court should also disregard Defendant's attempt to rely on extrinsic evidence to read a "physical" limitation into the claims. As an initial matter, Hugging Face presents no expert opinion supporting its arguments. (*See generally*, *supra*.) Further, the dictionary definitions to which Hugging Face points are inapposite. Specifically, Defendant cites to dictionary definitions for "cache memory," arguing that local, or cache, memory is defined to be physical. (*Supra* pp. 24-25.) But this argument ignores that the "cache" in "cache memory" in the context of the asserted patents refers to the "key *cache*" and "value *cache*" tensors. Indeed, FriendliAI explained this at length at in its opening brief, and Hugging Face did not dispute that "a POSITA would readily understand in light of the specification that the use of the word 'cache' in 'cache memory' refers to the KV cache." (*See supra* pp. 18-19 (citing Ex. 2 at claims 3 and 5, 23:4-15, Figs. 2A2-B, 3A-3B, 5A-5D); *see also generally supra*) Defendant's attempt to misconstrue the claimed memory as a local or cache memory is inconsistent with the asserted patents. Additionally, Defendant argues that Newton's Telecom Dictionary, the dictionary cited by FriendliAI, "confirms 'memory' is a physical structure *as it is 'part of a computer*.'" (*Supra* p. 24 (emphasis added); *see also* Ex. 4 (*Memory*, Newton's Telecom Dictionary (30th ed. 2016).) But the mere fact that memory is "part of a computer" does not require it to be physical, as again, computers commonly employ non-physical memory, such as virtual memory. (*See* Ex. 3 (Expert Declaration) at 12).

Defendant's flawed construction reads out the term "dedicated." Indeed, Defendant makes no meaningful attempt to address this deficiency, instead merely arguing that any dispute on the

"dedicated" language "misses the bigger picture" and does not address the requirement for physical structure. (*See supra* p. 25.) As the Federal Circuit has instructed, it is improper to read out words from the claims, and the Court should decline Defendant's attempt to do so. *Bicon, Inc. v. Straumann Co.*, 441 F.3d 945, 951 (Fed. Cir. 2006) (explaining that to read limitations out of a claim would "be contrary to the principle that claim language should not [be] treated as meaningless").

In sum, the Court should reject Defendant's attempt to redefine "memory," in contravention of its plain and ordinary meaning, to require a physical structure. The Court should further reject Defendant's attempt to misconstrue "cache" and read out "dedicated." FriendliAI respectfully submits that the terms "dedicated cache memory"/ "cache memory…dedicated" are clear from the claims and specification and do not require further construction. To the extent the Court believe a construction is necessary, FriendliAI has proposed the alternative construction of "memory for storing an internal state for a request."[5]

4. Defendant's Surreply Position

FriendliAI argues generic "memory" can include non-physical, virtual memory and ignores the specific type of memory required by the claims—*i.e.*, "cache memory." *Supra* at 25-29. "Cache memory" requires a physical memory structure.

The plain claim language supports Hugging Face's proposed construction of "cache memory" because it describes the physical location of the cache memory relative to the

[5] Defendant argues that there is a "contradiction" between the claim language and FriendliAI's proposed definition of "cache memory," because the claims allow for storing a key cache tensor, while FriendliAI's proposed definition refers to "an internal state." (*Supra* pp. 24-25.) The Court should reject Defendant's argument. As explained in the opening brief, the key cache tensor and value cache tensor are simply an example of an internal state. (*Supra* pp. 18-19.) Defendant does not dispute or otherwise address this point, and there is thus no support for any argument that FriendliAI's construction is somehow inconsistent with the claim language.

execution engine. Specifically, the cache memory must be *in* the execution engine (Set 1), or the execution engine must *include* the cache memory (Set 2). *See* Response at 10 (discussing Set 1 and Set 2 claims). In contrast, the claims use words like "associated" to describe non-physical relationships. *See* Ex. 2, claim 1 ("one or more input tensors *associated* with the batch of requests"). The plain claim language describes the physical position of a specific type of physical "cache memory," notwithstanding the statements of FriendliAI's expert.[6]

FriendliAI further references example disclosures in the specification pertaining to physical memory, including Random Access Memory (hardware), a dynamic storage device (hardware), magnetic disc (hardware), optical disk (hardware), and solid-state memory device (hardware) that purportedly show the claimed "cache memory" can be non-physical. *Supra* at 26. But these examples all relate to hardware, *i.e.*, physical structures, and support Hugging Face's proposed construction.

FriendliAI next cites yet another extrinsic reference, here Lenovo's website, which states "***[p]hysical memory*** refers to the actual ***random access memory (RAM)*** installed in your computer." *See supra* at 26, n3.[7] Lenovo's description of "cache memory" makes clear that "at

[6] Contrary to FriendliAI's assertions, expert testimony is not required to resolve the subject indefiniteness issues because the patent is clear on its face and there are no allegations by FriendliAI that the claims contain any specialized terms of art. *See Meyer Intellectual Properties Ltd. v. Bodum, Inc*., 690 F.3d 1354, 1374 (Fed. Cir. 2012) ("It is well-established … that, where the technology involved is easily understandable, expert testimony is not required."); *see also Phillips v. AWH*, 415 F.3 d 1303, 1318 (Fed. Cir. 2005) (en banc) ("conclusory, unsupported assertions by experts as to the definition of a claim term are not useful to a court"); *Wyers v. Master Lock Co*., 616 F.3d 1231, 1239, 1242 (Fed. Cir. 2010) (legal determinations can rely on "logic, judgment, and common sense, in lieu of expert testimony.").

[7] *Supra* at 26, n3 (citing https://www.lenovo.com/us/en/glossary/physical-memory/) (Note, the Lenovo reference teaches "virtual memory" refers to a "reserved space" or partition of physical memory on the SSD).

its core, ***cache memory is a form of random access memory (RAM)***." [8] FriendliAI's own evidence expressly states cache memory is a form of ***physical*** RAM, which supports Hugging Face's proposed construction.

Turning to the word "dedicated," FriendliAI's arguments relate to data being stored in the cache memory, which includes key cache tensors and value cache tensors. *Supra* at 28. Hugging Face does not dispute the claimed cache memory stores tensor data, however, storing data does not change the fact that cache memory is a physical structure in the execution engine.

FriendliAI's remaining arguments ask the Court to disregard Hugging Face's multiple, consistent dictionary definitions for the claimed physical "cache memory."[9] *Supra* at 28; *see* Ex. 10, *Encyclopedia Britannica*, May 20, 2021 (cache memory); Ex. 11, *The Authoritative Dictionary of IEEE Standard Terms*, 7th ed. (cache memory); Ex. 12, *A Dictionary of Computer Science*, 7th ed. (cache memory); *see also* Response at 12-13. FriendliAI does not explain how its definition of "memory" is inconsistent with Hugging Face's proposed construction. *Supra at 28*; *see also supra* at 23-25.

The intrinsic record, as well as the cited extrinsic dictionaries, make clear that the claimed type of "cache memory" is a physical structure. Thus, the claimed "cache memory" requires "physical memory in, or a component part of, the execution engine." The Court should adopt Hugging Face's proposed construction accordingly.

B. "scheduler"

Claim Term	FriendliAI's Proposed Construction	Defendant's Proposed Construction

[8] https://www.lenovo.com/us/en/glossary/what-is-cache-memory/.

[9] Notably, FriendliAI provided a dictionary definition for generic "memory," but not "cache memory." *See supra* at 19-20 (citing Ex. 4, (*Memory*, Newton's Telecom Dictionary (30th ed. 2016)).

"scheduler"	Not indefinite. No additional construction necessary. In the alternative: "the component that schedules a batch of requests for execution on an execution engine."	Indefinite. Governed by 35 U.S.C. § 112(b).

1. Plaintiff's Opening Position

Defendant asserts that the term "scheduler" is indefinite under 35 U.S.C. § 112(b). The term "scheduler" is recited in independent claims 1 and 10 of the '775 Patent and independent claims 1, 11, 21, 28, 35, 43, 51, and 59 of the '520 Patent. FriendliAI respectfully submits that the claim term "scheduler" is not indefinite. Indeed, the plain meaning of the term is clear from the claims and specification, and therefore the term needs no construction. In the alternative, to the extent that the Court determines that the claim term requires construction, FriendliAI proposes that the Court construe the term in a manner that comports with the plain language of the asserted claims, the specification, and extrinsic evidence, as "the component that schedules a batch of requests for execution on an execution engine."

A POSITA would readily understand the meaning of "scheduler" in the context of the claims. For example, independent claim 1 of the '520 Patent recites "receiving, by a serving system, one or more requests for execution, the serving system including a ***scheduler*** and one or more execution engines …", "scheduling, ***by the scheduler***, a batch of requests including the one or more requests for execution on an execution engine . . .", and "scheduling, ***by the scheduler***, a second batch of requests for execution on the execution engine . . ." The claim language thus makes clear that the scheduler is a component of the serving system, which schedules batches of requests for execution on the execution engine. Because the plain and ordinary meaning is clear from the claim language, this term needs no construction. *Chef Am.,* 358 F.3d at 1375 ("[W]e have

repeatedly declined to rewrite unambiguous patent claim language"); *Mentor H/S,* 244 F.3d at 1380 (there is a "heavy presumption" in favor of ordinary meaning).

In addition to the clear claim language, the specification provides a clear description of the "scheduler" of the patented invention. For example, the specification explains that:

> In one embodiment, the serving system 435 includes a request processor 580 and a ***scheduler*** 585 each coupled to the one or more execution engines. The request processor 580 receives requests and forwards the requests to the ***scheduler*** 585 … The ***scheduler*** 585 receives the forwarded requests from the request processor 580 and maintains an incoming request queue for storing new requests to be processed. The ***scheduler*** 585 forms a batch of requests and ***schedules*** the batch for execution on an execution engine. In one embodiment, the ***scheduler*** 585 is configured to monitor which batch of requests were distributed to each execution engine and how many iterations of the transformer model have been performed for each request and whether the request has been completed. The ***scheduler*** 585 is also configured to monitor the available cache memory in each execution engine.

(Ex. 1 at 22:32-53.) As explained above, the scheduler is part of the serving system and is coupled to the one or more execution engines. (*Id*.) The scheduler receives the incoming requests from the request processor, and maintains a queue for storing new requests to be processed. (*Id*.) The scheduler forms a batch of requests, and schedules the batch for execution. (*Id*.) The scheduler also has the ability to monitor the available cache memory in each execution engine. (*Id*.) This architecture, and the claimed scheduler, is depicted in Figs. 5A-5D. (*Id.*) The specification provides additional examples of the processing of requests using the scheduler in Figs. 5A-5D. (Ex. 1 at 22:65-23:35, 24:7-22.)

Beyond the intrinsic evidence, extrinsic evidence confirms the meaning of "scheduler" is well-known in the field of computer science and a POSITA would understand the recited "scheduler" component. For example, *A Dictionary of Computing* describes a scheduler as "[t]he code responsible for controlling use of a shared resource…When used without further qualification, the word scheduler refers to controlling the use of the processors." (Ex. 5 (*scheduler*, *A Dictionary of Computing*, Oxford University Press (6th ed. 2008)).) Similarly, another

computing dictionary defines scheduler as "a program that organises the use of a CPU or of peripherals which are shared by several users." (Ex. 6 (*scheduler*, *Dictionary of Computing*, A&C Black Publishers (6th ed. 2010)).) Both these definitions define "scheduler" in a manner similar to the way the scheduler is recited in the claims, the specification, and how a POSITA would understand the term. In the most simple terms, a scheduler is a component that schedules tasks that use processors or other resources in a computer system. In the asserted patent, the processors are hardware accelerators associated with the execution engines. The hardware accelerators may be one or more Graphics Processing Units ("GPUs"), Tensor Processing Units ("TPUs"), or Central Processing Units ("CPUs"). (Ex. 1 at 5:1-36.) The scheduler in the asserted patents, like those defined by computing dictionaries, is the component that is responsible for deciding which requests are added to a batch and executed on the execution engine.

Defendant proposes that the claim language is not readily understandable and should be invalid as indefinite. But Defendant has articulated no reason why the term "scheduler" is indefinite, let alone shown by clear and convincing evidence that this readily understandable term is indefinite. *Ironburg Inventions Ltd. v. Valve Corp.*, 64 F.4th 1274, 1284 (Fed. Cir. 2023) ("any fact critical to a holding on indefiniteness ... must be proven by the challenger by clear and convincing evidence.") As explained above, based on the claim language and the specification, a POSITA would readily understand the meaning of this claim term, and would understand, with reasonable certainty, the scope of the "scheduler" of the patented invention. (Ex. 3 at ¶¶ 44-48.)

The claim language of the asserted patents, the specification of each patent, and the extrinsic evidence all show that the term "scheduler" is definite, and does not require any additional construction.

2. Defendant's Answering Position

The claimed "scheduler" is indefinite because the claims and specification do not provide

any reasonable degree of certainty regarding the metes and bounds of what the scheduler is (or is not). FriendliAI's proposed alternate construction exemplifies the issue. FriendliAI proposes to replace "scheduler" with a generic "component" that performs a function already recited in the claims. The table below provides a side-by-side comparison between FriendliAI's proposed construction and the relevant recital from representative claims 1:

Representative Claim 1 (Ex. 2 | '775 patent)	**FriendliAI's Proposed Construction**
scheduling, by the scheduler, a batch of requests including the one or more requests for execution on an execution engine	"the component that schedules a batch of requests for execution on an execution engine"

Replacing the word "scheduler" with the word "component" does not provide any clarity or certainty regarding the metes and bounds of the "scheduler." As discussed, the claimed invention recites distinct hardware components, including a "serving system" and an "execution engine," as well as some kind of "scheduler" to perform the recited batching processes. *See* Ex. 2, claim 1; Ex. 1, claim 1. The independent claims introduce the "scheduler" as a distinct element that performs specific functions, but the neither the claims nor the specification explain what the scheduler is.

The specification describes the serving system hardware and the execution engine hardware, but it provides no guidance regarding what the claimed "scheduler" is, or otherwise provide to any corresponding hardware structure. At best, Figs. 5A-5D illustrate a serving system 435 that includes a request processor 580 and a scheduler 585. Ex. 2 at Figs. 5A-5D. The specification explains the request processor 580 and scheduler 585 are "coupled to execution engines 590A and 590B," which implies the scheduler requires a physical structure for "coupling" to the execution engine hardware. *Id*. at 22:23-28, 3:8-12 ("The serving system may include a request processor and a scheduler each coupled to one or more execution engines.").

While FriendliAI correctly notes the specification describes various "processor" hardware corresponding to the claimed "request processor," including graphics processing units (GPUs), tensor processing units (TPUs), or central processing units (CPUs), it does not point to any description of the scheduler hardware. This is because the specification provides no such description of the claimed "scheduler."

FriendliAI's remaining arguments do not resolve the indefiniteness issue, as the provided dictionary definitions for the "scheduler" would define it in terms of pure software/computer code. *See* Ex. 5; Ex. 6.

The term "scheduler" cannot be understood from the claim language or the specification. The Court should thus find the term to be indefinite.

3. Plaintiff's Reply Position

Defendant provides no legally proper argument or evidence to support its contention that the term "scheduler" is indefinite. Indeed, Defendant provides no expert declaration, and fails to rebut the opinion of FriendliAI's expert, Dr. Pazzani. (*See generally, supra*; *see also supra* pp. 34-35.) Further, while FriendliAI's opening brief identifies ample disclosures from the specification and provides dictionary definitions that describe how a person of skill in the art would understand "scheduler," Defendant offers no dictionary definitions to support its indefiniteness argument, nor does it point to any inconsistencies in the patents that would somehow render the commonly-understood "scheduler" term indefinite.

Instead of putting forth evidence to support its position, Defendant merely argues, incorrectly, that the specification "provides no guidance regarding what the claimed 'scheduler' is, or otherwise provide to any corresponding hardware structure." (*Supra* pp. 35-36.) As an initial matter, Defendant's arguments are factually wrong: in addition to providing expert opinions and dictionary definitions, FriendliAI identified multiple portions of the asserted patents, including the

specification and claim language, that explain what the "scheduler" is. (*Supra* pp. 32-33.) For example, as outlined in FriendliAI's opening brief, the patents explain that the scheduler is part of the serving system and coupled to an execution engine, receives incoming requests, maintains a queue for storing requests to be processed, and schedules batches for execution. (*Supra* at 33 (citing Ex. 1 at 22:32-53).)

Moreover, Defendant improperly conflates indefiniteness with lack of written description when it (incorrectly) asserts that the specification contains no "description of the claimed 'scheduler.'" (*Supra* at 36.) Rather than addressing whether a person of skill in the art would understand the scope of "scheduler," Defendant instead based its argument on whether the term, as Defendants interpret it, finds support in the specification. Such an argument is based on the wrong legal standard and is therefore legally erroneous. *See Augme Techs., Inc. v. Yahoo! Inc.*, 755 F.3d 1326, 1340 (Fed. Cir. 2014) (rejecting argument that specification "only discloses that the ingest server receives digital content, not a unique identifier as required by the claims" as applying the "wrong legal standard, i.e., written description or enablement as opposed to indefiniteness."); *TQ Delta, LLC v. 2Wire, Inc.*, 373 F. Supp. 3d 509, 524 (D. Del. 2019) (rejecting indefiniteness argument that "there is no embodiment demonstrating how to reduce rather than how to eliminate the difference in configuration latency" as it applies the "wrong legal standard").[10]

In sum, Defendant's attorney arguments are insufficient to meet the "clear and convincing evidence" burden for a claim to be held invalid for indefiniteness. *Ironburg Inventions Ltd. v. Valve Corp.*, 64 F.4th 1274, 1284 (Fed. Cir. 2023) (affirming district court decision that the claims

[10] As explained above, Defendant's argument relies on the unsupported assertion that the scheduler must be a "distinct hardware component[]" (*Supra* p. 35), which is also wrong and unsupported.

are not indefinite, noting that "the record does not contain clear and convincing evidence"); *SentriLock, LLC v. Carrier Fire & Sec. Americas LLC*, C.A. No. 20-520 (MN), 2024 WL 3328495, at *4 (D. Del. July 8, 2024) (declining to find claims are indefinite where defendant "rel[ies] solely on attorney argument" and the record lacks "clear and convincing evidence" of indefiniteness). Because Defendant has not provided any evidence showing indefiniteness or even articulated why a person of skill would not understand the "scheduler" term, the Court should reject Defendant's indefiniteness arguments. "Scheduler" is a commonly understood term that does not require any construction. To the extent the Court finds that a construction for "scheduler" would be helpful, FriendliAI respectfully requests that the Court construe the term as "the component that schedules a batch of requests for execution on an execution engine."

4. Defendant's Surreply Position

The dispute regarding "scheduler" is directed to *what* the scheduler is (or is not), not *how* the scheduler functions. FriendliAI focuses solely on *how* the scheduler functions. *supra* at 36-38. When considered from the proper perspective of what the scheduler is, the term is indefinite. *See Nautilus, Inc. v. Biosig Instruments, Inc*., 572 U.S. 898, 901 (2014) (A term is indefinite where the claims, "read in light of the specification delineating the patent, and the prosecution history, fail to inform, with reasonable certainty, those skilled in the art about the scope of the invention."). Put simply, the independent claims introduce the "scheduler" as a distinct element, but the neither the claims nor the specification explain its objective boundaries with sufficient clarity to inform those skilled in the art.

The parties agree the "scheduler" is part of the serving system and coupled to an execution engine. *See supra* at 36-37. But FriendliAI does not identify which part of the serving system corresponds to the scheduler, or how the scheduler is coupled to the execution engine hardware. *Supra* at 35-36. The asserted claims inherently rely on these structural relationships,

but do not disclose what the scheduler components are or what hardware structures enable the claimed functionality.

Turning to the specification, the execution engine hardware is described, but there is no mention or discussion of any hardware corresponding to the scheduler. *See*, *e.g.*, Ex. 2 at 4:60-64 (the execution engines are "built on specialized hardware accelerators such as graphics processing units (GPU's) or tensor processing units (TPU's)"; 5:10-12 ("[e]ach execution engine may include a set of cores (e.g., GPU cores) coupled to local memory (e.g., GPU memory), and may be composed of one or more hardware accelerators.") As the specification fails to provide any description of what the scheduler is, it does not provide any objective boundaries for those skilled in the art. *Interval Licensing LLC v. AOL, Inc.*, 766 F.3d 1364, 1371 (Fed. Cir. 2014) ("The claims, when read in light of the specification and the prosecution history, must provide objective boundaries for those of skill in the art.").

FriendliAI's extrinsic dictionary definition only exacerbates the issue. *See supra* at 34 (citing Ex. 5). This definition defines "scheduler" in terms of pure software and computer code. *Id*. At the same time, the parties agree the "scheduler" is coupled to the execution engine—*e.g.*, a hardware component. *See supra* at 33. FriendliAI does not explain how a scheduler consisting of software code can "couple" to the execution engine hardware.

At bottom, the term "scheduler" cannot be understood from the claim language or the specification with any reasonable degree of certainty by a person of ordinary skill in the art. It is therefore indefinite.

C. "at least one batch operation"

Claim Term	FriendliAI's Proposed Construction	Defendant's Proposed Construction
"at least one batch operation"	Not indefinite. No additional construction necessary.	Indefinite. Governed by 35 U.S.C. § 112(b).

	In the alternative: "at least one operation of the transformer model that is applied to a batch."	

1. Plaintiff's Opening Position

Defendant asserts that the term "at least one batch operation" is indefinite under 35 U.S.C. § 112(b). The term "at least one batch operation" is recited in independent claims 1 and 10 of the '775 Patent and claims 1, 11, 21, 28, 35, 43, 51, and 59 of the '520 Patent. FriendliAI respectfully submits that the claim term "at least one batch operation" is not indefinite and does not need additional construction. In the alternative, to the extent that the Court determines that the claim term requires construction, FriendliAI proposes the construction of "at least one operation of the transformer model that is applied to a batch," which is consistent with the plain language of the asserted claims and the specification.

The meaning of the claim term "at least one batch operation" is clear from the claim language itself. Each independent claim recites "wherein applying the transformer model comprises applying at least one batch operation to one or more input tensors associated with the batch of requests." (Ex. 1, cl. 1, 11, 21, 28, 35, 43, 51, 59; Ex. 2, cl. 1, 10.) As evident from the plain language of the claims, a batch operation is simply executing a transformer model operation on a batch of requests. A POSITA would readily understand the meaning of this term in the context of the claims. (Ex. 3 at ¶¶ 53-55.)

Moreover, the specification provides a clear description of the "at least one batch operation" in the context of the Asserted Patents. Specifically, the specification explains that "the ***operations of the transformer model*** 200 are configured as ***batch operations*** in which data for a ***batch of requests are processed together***." (Ex. 1 at 7:43-45 (emphasis added). The claimed operations are the "operations of the transformer model," and as depicted in Fig. 2A, for example,

such operations may include the "QKV Operation 215," "Split 220" operation, "Self-Attention 225" operation, and "Attention Linear 230" operation are all batch operations. (*See id.*; *see also* Ex. 1, Figs. 2A-2B, 8:21-26 (explaining with reference to the embodiment in Fig. 2A, that "each of the blocks in the decoder are performed on an execution engine ***as a batch operation***" (emphasis added).) Further, such operations are "batch" operations in that they are applied to a batch of requests that are processed together, as opposed to individually. (*See* Ex. 1 at 7:43-45; *see also id.* at 1:45-49 ("In one instance, the inference system ***processes requests in batches*** to achieve high processor utilization on the accelerators. Specifically, the inference system may ***process multiple requests in a batch together*** to exploit the amount of parallel computation units in the hardware accelerators." (emphasis added)).) A POSITA would read the specification and understand what a batch operation is in the context of the patented invention. (Ex. 3 at ¶¶ 53-54.)

Once again, Defendant proposes that the claim term should be invalid as indefinite. But Defendant has not articulated why this claim is indefinite, let alone shown by clear and convincing evidence that this readily understandable term is indefinite. As explained above, the meaning and scope of the term "at least one batch operation," in light of the claims and specification, would be clear to a POSITA. (Ex. 3 at ¶¶ 53-56.)

Accordingly, FriendliAI submits that the specification and the claim language of the asserted patents show that the term "at least one batch operation" is definite and does not require any additional construction. Should the Court believe a construction would be helpful to the jury, FriendliAI proposes that the Court construe the term consistent with its plain and ordinary meaning, as "at least one operation of the transformer model that is applied to a batch."

2. Defendant's Answering Position

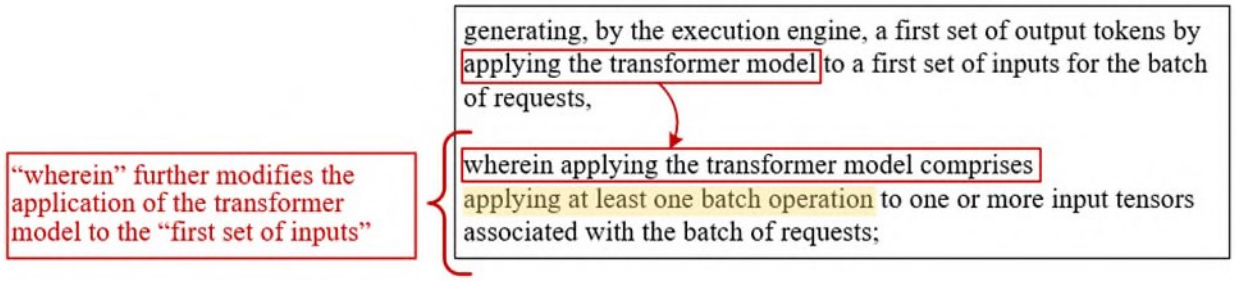
As used in the claims, "at least one batch operation" is indefinite. The indefiniteness arises from an incurable disconnect regarding the application of "one batch operation" to "one"

(*i.e.*, singular) input tensor. "[A]t least one batch operation" appears in the following recital of representative claim 1 of the asserted patents:

> "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*"

Ex. 2, claim 1 (emphasis added); Ex. 1, claim 1 (emphasis added).

As illustrated below, representative claim 1 includes a nested "wherein" clause that introduces the "at least one batch operation" recital to modify the execution engine's application of the transformer model to "a first set of inputs (*i.e.*, *plural*) for the batch of requests":



Ex. 2, claim 1 (annotated); Ex. 1, claim 1 (annotated).

The adjective "batch" describes the type of "operation." The surrounding claim language makes it clear the claimed "batch operation" refers to a type of grouping action for combining multiple inputs together. For example, the claim initially states the execution engine "appl[ies] the transformer model" to a "set" or group of inputs (*i.e.*, plural), and the modifying "wherein" language further explains the transformer model itself applies the "batch operation" to group or combine multiple "input tensors" (*i.e.*, plural) together. *Id.*

However, the "batch operation" recital includes an internal inconsistency that renders it indefinite. The recital states the "batch operation" can somehow be applied to "one" (*i.e.*, singular) input tensor associated with the batch of requests. The surrounding claim language does not provide any guidance to explain how the "batch operation" can group or batch "one,"

single input tensor by itself, nor does it make any sense to do so. Thus, the "batch operation" claim language is indefinite.

The intrinsic evidence further confirms this. The parties do not dispute the claimed "batch operation" term does not have any special meaning. In the context of the claims, the "at least one batch operation" refers to the selective batching process. *See* Ex. 2 at Figs. 3A-3B. As discussed, FriendliAI amended the claims during prosecution and expressly stated the claimed invention requires both the selective batching and dynamic batching processes. *See* Ex. 7, 189; Ex. 8, 284. The specification describes the claimed "at least one batch operation" in the context of selective batching. For example, consistent with the claim language, the specification explains that when the execution engine applies the transformer model to a set of inputs, the transformer model further applies a "batch operation" that concatenates multiple inputs (*e.g.*, sequences X_1, X_2, X_3) together for certain operations. *Id*. at Figs. 3A-3B, 11:19-24, 11:54-56. Fig. 3A further illustrates the transformer model applying the claimed "batch operation" to combine or "batch" multiple inputs (X_1, X_2, X_3) together for specific processing steps:

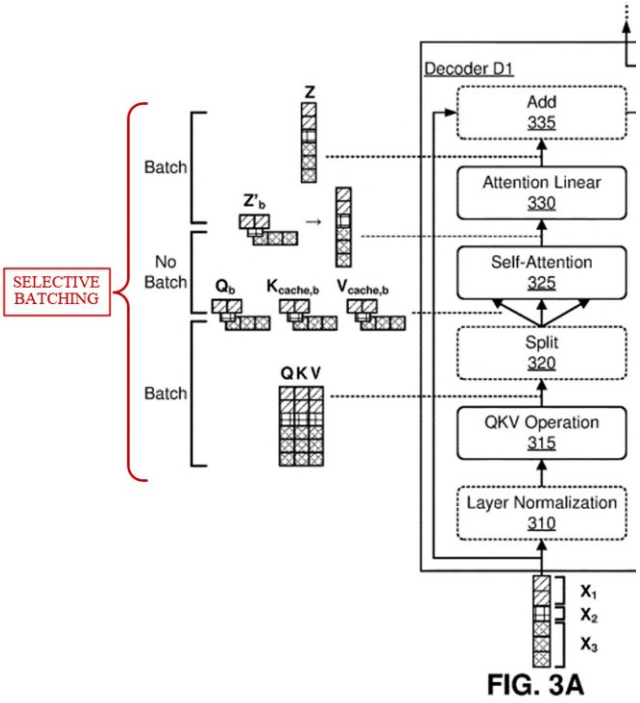


FIG. 3A

'775 patent, Fig. 3A
(annotated)

Importantly, the description of the selective batching process consistently describes the "batch operation" as grouping or concatenating multiple inputs together, which is different from processing non-batched, individual inputs. *See id*. at 11:26-22:21, 11:41-44, Figs. 3A-3B (showing "a subset of operations in transformer model 300, specifically operations in the self-attention operation block 325, are executed separately instead of being processed as a batch operation"), 11:19-24 ("By selective batching, the inference system 130 can allow batching operations to be performed for a batch of requests with variable input or target length or internal state length to utilize the parallel computation capabilities of hardware accelerators."), 11:54-56.

On the other hand, FriendliAI argues "batch operation" can include any transformer model operation performed on a batch of requests. *Supra* at 40 ("a batch operation is simply executing a transformer model operation on a batch of requests"). This argument is contrary to the claim language and reads out "batch" from "batch operation," thereby redefining "batch

operation" to include any transformer model operation. *Id*. FriendliAI's arguments are incorrect for multiple reasons.

First, the claims already recite "applying the transformer model to a first set of inputs for the batch of requests," which requires performing transformer model operations on a batch of requests. FriendliAI's proposed construction would render the clause "applying at least one batch operation" meaningless. *See Nautilus*, 572 U.S. at 901.

Second, FriendliAI's proposed construction is incorrect because the claim languages uses the adjective "batch" to describe the type of "operation" (*e.g.*, a grouping action for multiple inputs). The claimed "batch operation" does not refer to any and all transformer model operations.

Third, FriendliAI's proposed construction relies on the embodiments illustrated in Figures 2A-2B, which do not relate to the claimed invention. *See supra* at 40-41. The batching process shown in Figs. 2A-2B requires input token sequences having the same length. Ex. 2 at 10:43-50. FriendliAI specifically amended the claims during prosecution to expressly require that "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. 7 at 179-193 (May 3, 2022 Amendment); Ex. 8 at 261-286 (Aug. 7, 2023 Amendment); *see also* Ex. 2 at 10:43-50 ("the methods of batching transformer models, such as the method described in conjunction with FIGS. 2A-2B, require that the length of input tensors for the batch be the same"). Moreover, FriendliAI expressly stated the claimed invention requires both selective batching—Figures 3A-3B—and dynamic batching—Figures 5A-5D—which is different from the batching process shown in Figures 2A-2B. *See* Ex. 7 at 189 (May 3, 2022 Amendment); Ex. 8 at 284 (Aug. 7, 2023 Amendment); *see also* Ex. 2 at 11:28-31 (the selective batching process shown in Figs. 3A-

3B is "different from FIG. 2A"); *see also PPC Broadband, Inc. v. Corning Optical Communications RF, LLC*, 815 F.3d 747, 755 (Fed. Cir. 2016) (rejecting the proposition that "each and every claim ought to be interpreted to cover each and every embodiment"); *Baran v. Med. Device Technologies, Inc.*, 616 F.3d 1309, 1316 (Fed. Cir. 2010) ("It is not necessary that each claim read on every embodiment."); *TIP Systems, LLC v. Phillips & Brooks/Gladwin, Inc.*, 529 F.3d 1364, 1373 (Fed. Cir. 2008) ("Our precedent is replete with examples of subject matter that is included in the specification, but is not claimed.").

FriendliAI's construction of "batch operation" does not cure the internal inconsistency found in the claim language. The "batch operation" recital requires an operation for grouping or batching multiple inputs together. But the claim language contemplates applying the "batch operation" to a non-batched, individual input tensor. Thus, the plain claim language setting forth the "batch operation" term is indefinite.

3. Plaintiff's Reply Position

Defendant bases its indefiniteness argument for "at least one batch operation" on a fundamental misunderstanding of term "batch operation." Specifically, Defendant claims that a "'batch operation' refers to a type of grouping action for combining multiple inputs together." (*Supra* pp. 42-43.) It then relies on this incorrect interpretation to assert an "internal inconsistency," because the claims contemplate applying a batch operation to "one" input tensor. (*Supra* pp. 42-43.) But Defendant's argument rests on an incorrect assumption—i.e., that a "batch operation" refers to a "grouping action."

As an initial matter, Defendant's interpretation of "batch operation" improperly ignores all descriptions of "batch operations" in the specification and figures. Examples of such descriptions were provided in FriendliAI's opening brief (*supra* pp. 42-44), and FriendliAI expounds upon such

descriptions here. The patent specification makes clear that a "batch operation" refers not to a process for grouping or batching inputs, but rather to a process performed *on a batch* of requests.

For example, the specification unambiguously states that "batch operations" are operations of the transformer model "in which data for a batch of requests are processed together":

> In one embodiment, ***the operations of the transformer model 200 are configured as batch operations in which data for a batch of requests are processed together.*** A batch operation is coupled to receive one or more input tensors that are each concatenations of input data ***for multiple requests in a batch***. The batch operation generates one or more output tensors that are each concatenations of output data ***for the multiple requests in the batch*** by applying the respective operation to the one or more input tensors. For a batch operation, an input tensor may be a concatenation of the input tensor for each respective request in the batch. Thus, the input tensor for the batch operation is a single, larger tensor that coalesces the input tensors for each request in the batch across an additional batch dimension.

(Ex. 2 at 7:35-48 (emphasis added).) As shown above, the specification not only defines "batch operations" as operations "in which data for a batch of requests ***are processed together***," but it also repeatedly refers to the "batch operation" as performing activity "*for* multiple requests in a batch." (*Id.* (emphasis added).) Notably, the specification does not refer to the "batch operation" as "combining," "grouping" or "concatenating" inputs or requests. Rather, the specification makes clear that the requests are already batched together before the "batch operation" is performed, and the "batch operation" functions to process the data for the batch of requests together. (*Id.*) Defendant's only argument in response to the above excerpt is that it applies to "embodiments illustrated in Figures 2A-2B," which Defendant contends is unrelated to the claimed invention. But Defendant provides no support for the unfounded notion that "batch operation" has a different meaning in the different embodiments of the patent.

Moreover, the specification distinguishes a batch operation from operations that are "executed separately":

> However, different from the transformer model 200 of FIGS. 2A-2B, ***a subset of operations in the transformer model 300***, specifically operations in the self-attention operation block 325, ***are executed separately instead of being processed as a batch operation***. As shown in FIG. 3A, the transformer model 300 includes a first layer normalization block 310, a QKV operation block 315, a split block 320, a self-attention block 325, an attention linear block 330, a first add block 335, a second layer normalization block 340, a first MLP block 345, a GeLU block 350, a second MLP block 355, and a second add block 360.

(Ex. 2 at 11:43-53 (emphasis added).) As shown above, the specification explains that each of the transformer operation blocks in Figs. 3A can be performed on the batch of requests—i.e., "as a batch operation"—instead of having to execute each request in the batch separately. (*Id*.) In other words, unlike operations that execute on or process requests "separately," batch operations are those that execute on, or process, a *batch* of requests. Again, the above portion of the specification makes no reference to batch operations that "group," "combine," or "concatenate" requests, as Defendant proposes.

Moreover, the examples of "batch operations" provided in the patent make clear that such operations do not refer to grouping or concatenating inputs. For example, the specification explains that the "attention linear block 330" (which does not refer to grouping or concatenating inputs) can be executed as a "batch operation,":

> ***The attention linear block 330 is coupled to receive the attention output tensor as the input tensor*** and generates an output tensor by applying an attention weight tensor that is a trained set of parameters of the transformer model 300 to the attention output tensor. The attention linear block 330 may be configured as one or more neural network layers. ***The attention linear block 330 may be executed as a batch operation.***

(Ex. 2 at 15:23-30 (emphasis added).) Likewise, Figure 3A, also cited in Defendant's Response brief, refers to various other operations—such as the "QKV Operation"—as "Batch" operations:

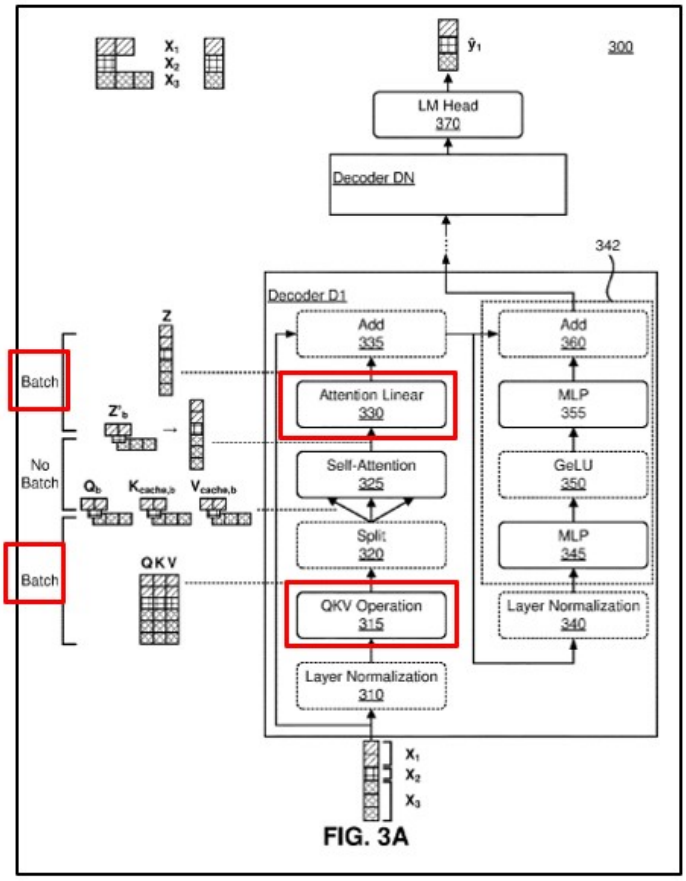


FIG. 3A

(Ex. 2 at Fig. 3A (annotated); *see also supra* p. 44 (citing Fig. 3A.)) As Figure 3A reflects, various operations of the transformer model—notably, *not* operations of combining or concatenating inputs—can be performed as (and therefore are) "batch" operations.

Defendant ignores *all* of these disclosures to claim that the "batch operation" is simply the concatenation or the "grouping action for combining multiple inputs together." (*Supra* pp. 42-43.) Yet Defendant provides no support from the specification for its interpretation—none of the excerpts that Defendant cites in its response brief refer to the batch operation as grouping or concatenating inputs. (*See generally*, *supra*.) As explained above and in FriendliAI's opening brief, a "batch operation" is an operation of a transformer model that is applied to a batch. [redacted]

[redacted]

Because the claimed "batch operation" is not a "grouping" operation as Defendant proposes, Defendant's indefiniteness argument necessarily fails. That is, Defendant argues that *if* "batch operation" is construed to mean "grouping," then the claims are indefinite because the claims recite applying a batch operation to a single input tensor, and it would be "internally inconsistent" to perform "grouping" on a *single* input tensor. As explained above, however, "batch operation" does not in fact refer to grouping, but rather refers to a transformer model operation applied to a batch of requests. Such a batch operation can readily be applied to "one or more input tensors associated with the batch of requests." There is no "internal inconsistency" in applying that *transformer model operation* to a single input tensor requests, and indeed Defendant does not argue otherwise.[11]

The term "batch operation" is clear from the specification and claims, and is described in significant detail in the patents. Defendant purposefully asserts a nonsensical construction of the term in order to assert that there is an internal inconsistency in the claims. When "batch operation" is understood according to the plain and ordinary meaning, there is no internal inconsistency in applying a batch operation to a single input tensor. The term "batch operation" therefore is not indefinite.

[11] Moreover, the specification makes abundantly clear that taking a transformer model operation for a batch of requests, and applying it to a single input tensor, creates no indefiniteness issues. Indeed, the specification even explicitly describes applying a batch operation to a single input tensor. (*See* Ex. 2 at Fig. 6B (step 616 describing the application of a "batch operation" to "the concatenated tensor" (singular).)

4. Defendant's Surreply Position

The parties conflicting interpretations of the "at least one batch operation" term underly the indefinite issue.

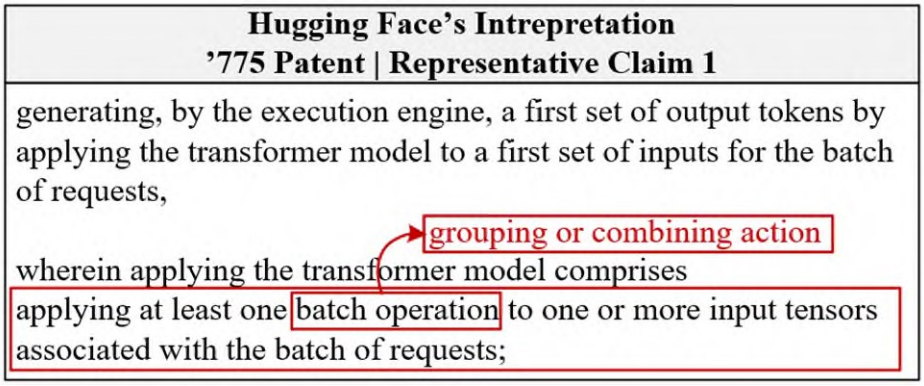
FriendliAI proposes to wholesale delete the adjective "batch" and rewrite the recital such that claimed "batch operation" refers to any possible transformer model operation "that is applied to a batch." *See* D.I. 109 at 5 (FriendliAI's proposed construction is "at least one operation of the transformer model that is applied to a batch"). The annotations below to representative claim 1 of the '775 patent illustrate FriendliAI's proposed construction.

FriendliAI's Proposed Construction '775 Patent | Representative Claim 1
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~~;

'775 Patent, representative claim 1
(annotated by Hugging Face to show FriendliAI's proposed construction).

As shown above, FriendliAI's proposed construction rewrites the recital contrary to the straightforward language. FriendliAI's construction requires the transformer model to apply any "one" operation "to the batch" itself, while the actual claim language states the transformer model applies the "batch operation to one or more input tensors associated with the batch."

In contrast, Hugging Face's interpretation is consistent with the plain claim language as its written—namely, the adjective "batch" describes the type of operation—*e.g.*, a grouping or combining action—applied to "one or more input tensors." The annotations below to representative claim 1 of the '775 patent below illustrate Hugging Face's proposed construction.



'775 Patent, representative claim 1
(annotated by Hugging Face to show Hugging Face's proposed construction).

Hugging Face's interpretation is also consistent with the prosecution history. Importantly, FriendliAI does not address its express statements to the Examiner during prosecution, which demonstrate the claimed "batch operation" refers to the selective batching disclosures (Figures 3A-3B) and requires grouping or combining multiple inputs together. *See Markman v. Westview Instruments, Inc*., 52 F.3d 967,980 (Fed. Cir. 1995) (en banc), *aff'd*, 517 U.S. 370 (1996). To overcome the prior art, FriendliAI informed the USPTO that the claimed invention requires both (1) selective batching (Figures 3A-3B) performed by the transformer model, and (2) dynamic batching (Figures 5A-5D) performed by the scheduler. Ex. 7 at 189 ('775 patent File History); Ex. 8 at 284 ('520 patent File History); *see Fenner Investments, Ltd. v. Cellco P'ship*, 778 F.3d 1320, 1323 (Fed. Cir. 2015) (quoting *Retractable Technologies, Inc. v. Becton, Dickinson & Co*., 653 F.3d 1296, 1305 (Fed. Cir. 2011)) ("Any explanation, elaboration, or qualification presented by the inventor during patent examination is relevant, for the role of claim construction is to 'capture the scope of the actual invention' that is disclosed, described, and patented."). This is inapposite to FriendliAI's proposed construction, which relies on disclosures that do not relate to the claimed selective "batch operation." *See supra* at 46 (citing *supra* at 42-44 (referencing disclosures corresponding to Figures 2A-2B)).

At the same time, turning to the Figures, FriendliAI does not dispute that the claimed

"batch operation" refers to selective batching operations performed by the transformer model 300 shown in Figures 3A-3B and the specification teaches selective batching is different than the batching shown in Figures 2A-2B. *See* Ex. 2 at 11:28-31, 11:41-46; *see also supra* at 46-49. The specification goes on to teach that "[s]pecifically, in the selective batching method, the inference system 130 may concatenate the input token sequences X_1, X_2, X_3 into a concatenated input tensor. Different from the batching method in FIGS. 2A-2B, the inference system 130 concatenates the input token sequences such that individual input tokens for the batch of requests are concatenated across one dimension, for example, the vertical dimension in FIGS. 3A-3B." Ex. 2 at 11:54-61. The foregoing disclosures contradict FriendliAI's statement that the specification does not describe selective batching as a grouping or concatenating function. *See supra* at 47.

Moreover, FriendliAI's argument that the selective batching process can include separately executing inputs for certain operations does not contradict Hugging Face's interpretation. *See supra* at 48-49. While selective batching can include operations by the transformer model to un-batch or separate the inputs for certain operations, this does not change the fact that the claimed selective "batch operation" refers to the grouping (or re-grouping) of multiple inputs for other operations. *See* Ex. 2 at 11:54-61 ("Specifically, in the selective batching method, the inference system 130 may concatenate the input token sequences X_1, X_2, X_3 into a concatenated input tensor."), Fig. 3A (transformer model 300 performs "Batch" operations for grouping inputs together and "No Batch" operations for separating the inputs for specific operations). FriendliAI's annotated version of Figure 3A also illustrates this point. *Supra* at 49 (annotating the transformer model performing the selective "Batch" operation to group multiple inputs together). When the transformer model performs the claimed "batch operation,"

it selectively batches inputs together for specific subsets of operations. Ex. 2 at 11:31-40, 11:12-17, 11:41-47, 11:41-47, Figures 3A-3B. FriendliAI's summary conclusions that the claimed selective "batch operation" refers to any transformer model operation being applied "to the batch" is unsupported by the intrinsic record.

Finally, FriendliAI's proposed construction represents an infringement-based construction derived from the deposition testimony of a Hugging Face witness specifically pertaining to the accused Hugging Face functionality, not the asserted patents. *See supra* at 49-50. However, "[a] claim is construed in the light of the claim language, the other claims, the prior art, the prosecution history, and the specification, *not* in light of the accused device." *Sprint Communications Co. L.P. v. Cox Communications Inc*., 302 F. Supp. 3d 597, 609 (D. Del. 2017) (quoting *SRI Int'l v. Matsushita Elec. Corp. of Am*., 775 F.2d 1107, 1118 (Fed. Cir. 1985) (en banc)) ("[C]laims are not construed 'to cover' or 'not to cover' the accused device. That procedure would make infringement a matter of judicial whim. It is only *after* the claims have been *construed without reference to the accused device* that the claims, as so construed, are applied to the accused device to determine infringement.").

In sum, the claimed "batch operation" requires a grouping or batching operation for combining *multiple* inputs together. However, in the context of the full recital, the claim language recites applying the "batch operation" to a "*one*" (single), non-batched input tensor, which results in an irreconcilable inconsistency. The claimed "batch operation" cannot be applied to "one" individual input tensor and thus, the "batch operation" recital is indefinite.

VII. CONCLUSION

Plaintiff's Opening

FriendliAI respectfully submits that each of the disputed claim terms do not require construction and should be interpreted according to their plain and ordinary meaning. FriendliAI

further submits that both "scheduler" and "at least one batch operation" are definite.

Defendant's Answering

For the foregoing reasons, Hugging Face requests that its proposed constructions be adopted, that FriendliAI's proposals be denied, and all other relief to which it may be entitled.

Plaintiff's Reply

For the reasons above, the term "dedicated cache memory"/ "cache memory … dedicated" should be construed according to its plain and ordinary meaning. Additionally, Defendant has *not* met its burden to prove by clear and convincing evidence that the terms "scheduler" and "at least one batch operation" are indefinite.

Defendant's Surreply

For the foregoing reasons, Hugging Face requests that its proposed constructions be adopted, that FriendliAI's proposals be denied, and all other relief to which it may be entitled.

Respectfully submitted,

POTTER ANDERSON & CORROON LLP

By: */s/ David E. Moore*

David E. Moore (#3983)
Bindu A. Palapura (#5370)
Andrew L. Brown (#6766)
Hercules Plaza, 6th Floor
1313 N. Market Street
Wilmington, DE 19801
Tel: (302) 984-6000
dmoore@potteranderson.com
bpalapura@potteranderson.com
abrown@potteranderson.com

Attorneys for Plaintiff FriendliAI Inc.

OF COUNSEL:

Michael J. Sacksteder
Shreyas A. Kale
Samantha Ong
FENWICK & WEST LLP
555 California Street, 12th Floor
San Francisco, CA 94104
Tel: (415) 875-2300

Jessica Kaempf
FENWICK & WEST LLP
401 Union Street, 5th Floor
Seattle, WA 98101
Tel: (206) 389-4550

POLSINELLI PC

/s/ Stephen J. Kraftschik
Stephen J. Kraftschik (#5623)
222 Delaware Avenue, Suite 1101
Wilmington, DE 19801
(302) 252-0920
skraftschik@polsinelli.com
Attorneys for Defendant Hugging Face, Inc.

OF COUNSEL:

Jason A. Wietjes
POLSINELLI PC
2950 N. Harwood St., Ste. 2100
Dallas, TX 75201
(214) 661-5519
jwietjes@polsinelli.com

Adam P. Daniels
POLSINELLI LLP
2049 Century Park E.
Los Angeles, CA 90405, Ste. 2900
(310) 556-6754
adaniels@polsinelli.com

Jahnathan L.D. Braquet
POLSINELLI PC
1000 Louisiana Street, Suite 6400
Houston, TX 77002
(713) 374-1600
jbraquet@polsinelli.com

Dated: October 11, 2024
Public Version Dated: October 17, 2024