IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

FRIENDLIAI INC.,	)	
	)	
Plaintiff,	)	
	)	C.A. No.
v.	)	
	)	**JURY TRIAL DEMANDED**
HUGGING FACE, INC.,	)	
	)	
Defendants.	)	
	)	

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff FriendliAI Inc. (“FriendliAI”), for its Complaint against Defendant Hugging Face, Inc. (“Hugging Face” or “Defendant”), hereby alleges as follows:

NATURE OF THE ACTION

1. This is a civil action arising under the Patent Laws of the United States, 35 U.S.C. §§ 271 *et seq*., for infringement of United States Patent No. 11,442,775 (the “’775 patent” or the “Patent-in-Suit”) relating to artificial intelligence technology, specifically for machine-learning transformer neural network models.

2. Artificial intelligence, or AI, is a field of computer science that focuses on creating intelligent machines capable of performing tasks that typically require human intelligence. Using algorithms, data, and computational power, AI can understand, process, and generate human language; analyze and interpret data; and make decisions or take actions to achieve specific goals. AI has a wide range of applications across various industries, including healthcare, finance, transportation, manufacturing, cybersecurity, gaming, and entertainment. The global AI market size is projected to grow at a Compound Annual Growth Rate (CAGR) of 36.8%, reaching $1,345.2 billion by 2030 from $150.2 billion in 2023. *See*

https://www.marketsandmarkets.com/PressReleases/artificial-intelligence.asp.

3. FriendliAI is a pioneering AI company that focuses on large-scale generative AI technology. Following substantial research, FriendliAI developed a serving engine for large-scale generative AI models that optimizes efficiency, throughput and latency. FriendliAI's novel serving engine can be used for a variety types of generative AI tasks, including data generation, translation, sentiment analysis, text summarization, auto-correction and the like. Such efforts by FriendliAI, and its founder and CEO Dr. Byung-gon Chun, have resulted in the issuance of multiple patents, including the Patent-in-Suit.

THE PARTIES

4. Plaintiff FriendliAI is a company organized and existing under the laws of the Republic of Korea, with its principal place of business at 5F, AMC Tower, 222 Bongeunsa-ro, Gangnam-gu, Seoul, 06135, Korea.

5. Defendant Hugging Face, Inc. is a company organized and existing under the laws of the State of Delaware, with its principal place of business at 20 Jay St, Ste 620, Brooklyn, New York, 11201.

JURISDICTION AND VENUE

6. This action arises under the patent laws of the United States, including 35 U.S.C. §§ 271 *et seq*. The jurisdiction of this Court over the subject matter of this action is proper under 28 U.S.C. §§ 1331 and 1338(a).

7. Venue is proper in this District pursuant to 28 U.S.C. §§ 1391(b), (c), and 1400(b). Defendant is an entity organized under the laws of Delaware and resides in Delaware for purposes of venue under 28 U.S.C. § 1400(b). Defendant conducts business in Delaware, at least by offering for sale, selling, and otherwise making available products and services through its website,

which are accessible in Delaware. Defendant has also committed and continues to commit acts of infringement in this District.

8. This Court has personal jurisdiction over Defendant because Defendant conducts business in Delaware by at least offering for sale, selling, and otherwise making available products and services through its website, which are accessible in Delaware, and because infringement has occurred and continues to occur in Delaware.

9. Personal jurisdiction also exists over Defendant because it is an entity incorporated in and organized under the law of Delaware.

BACKGROUND OF THE PATENTED TECHNOLOGY

10. The founder and CEO of FriendliAI is Dr. Byung-gon Chun, a computer scientist and Professor known for his contributions to the field of computer systems, distributed computing, and artificial intelligence. Dr. Chun received his Ph.D. in Computer Science from the University of California, Berkeley. He is currently a Professor in the Computer Science and Engineering (CSE) Department at Seoul National University (SNU), where he leads the Software Platform Lab (SPL), conducts research on machine learning systems, and teaches courses, including courses on Artificial Intelligence and Big Data Systems. Dr. Chun has published numerous papers in reputable conferences and journals, and has received a number of awards, including the EuroSys 2021 Test of Time Award, the 2020 ACM SIGOPS Hall of Fame Award, the 2020 Google Research Award, the 2019 SNU Education Award, and the 2018 Amazon Machine Learning Research Award. Dr. Chun's work has contributed to advancements in the design and optimization of the performance, scalability, and reliability of large-scale distributed systems, including serving and training transformer-based generative AI models.

11. The vision of FriendliAI is to enable innovation by lowering barriers to serving generative AI. In furtherance of that vision, FriendliAI developed PeriFlow (a version of a

distributed serving system called Orca), a patented solution that efficiently serves large-scale AI transformer models. PeriFlow/Orca uses a novel optimization technique referred to as batching with iteration-level scheduling, also known as dynamic batching or continuous batching, which provides for improved throughput and decreased latency as compared to prior art systems.

12. Before FriendliAI's inventions, transformer models had begun to be widely used for AI applications. Pre-existing transformer models, however, suffered from latency and throughput issues, especially when used for large-scale applications. Dr. Chun and his colleagues at FriendliAI and Seoul National University recognized that such issues resulted from the use of a blunt scheduling mechanism, which was primarily designed to schedule executions at request granularity. In existing models, a form of "batching," referred to as naive batching or static batching, could be used to process multiple requests at once, in order to increase overall throughput. But because requests can vary in complexity and length, particularly in generative AI applications, and because the transformer generation model only returns the execution results to the serving system when it finishes processing all requests in the batch, a request that "finishes" early cannot be sent to the client immediately, but rather must wait until the last request in the batch is "finished," imposing a substantial amount of extra latency (or in plain language, causing delays in processing the requests in the batch). Similarly, when a new request arrives in the middle of the current batch's execution, the aforementioned scheduling mechanism makes the newly-arrived request wait until all requests in the current batch have finished. The inflexibility of such scheduling mechanisms resulted in high latency and low overall throughput.

13. Dr. Chun and his colleagues recognized that a method of scheduling the system on a finer granularity could resolve the latency and throughput issues associated with existing systems.

14. After substantial research, Dr. Chun developed novel technology referred to as batching with iteration-level scheduling (also called dynamic batching or continuous batching). Iteration-level scheduling allows for a finished request to be sent to a client, and for new requests to be sent to the execution engine, before all requests in a batch are completed. Iteration-level scheduling provides for a highly efficient and scalable serving of generative AI transformer models, with optimized throughput and latency.

15. Batching with iteration-level scheduling is described in a paper co-authored by Dr. Chun and others at FriendliAI, and presented in July 2022, during the Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation. *See* Ex. 3 (Yu, G.-I., Jeong, J.S., Kim, G.-W., Kim,S., and Chun, B.-G. Orca: A distributed serving system for {Transformer-Based} generative models. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pp. 521–538, 2022).

16. Dr. Chun's paper is recognized by others in the industry as the "first" to disclose iteration-level scheduling (also known as dynamic batching or continuous batching). *See* Ex. 4 (https://www.anyscale.com/blog/continuous-batching-llm-inference).

17. The industry has also recognized the benefit of FriendliAI's novel technology—an increase in throughput and decrease in latency—with one article detailing how "continuous batching" (which is also known in the industry as dynamic batching, or batching with iteration-level scheduling) "improves throughput by wasting fewer opportunities to schedule new requests, and improves latency by being capable of immediately injecting new requests into the compute stream." *See* Ex. 4 (https://www.anyscale.com/blog/continuous-batching-llm-inference).

18. The Patent-in-Suit resulted from Dr. Chun's research to develop an innovative serving engine for generative AI transformer models.

PATENT-IN-SUIT

19. The ’775 Patent, entitled “Dynamic Batching for Inference System for Transformer-Based Generation Tasks” was duly and legally issued by the United States Patent and Trademark office on September 13, 2022. The inventors of the patent are Gyeongin Yu, Geon-Woo Kim, Joo Seong Jeong, Soojeong Kim and Byung-Gon Chun, and the patent is assigned to FriendliAI. A copy of the ‘775 Patent is attached hereto as Exhibit 1.

20. FriendliAI is the exclusive owner of all rights, title, and interest in the ’775 Patent, and has all rights to bring this suit to recover damages for any current or past infringement of the ‘775 Patent. *See* Ex. 1.

21. The ’775 Patent is directed to, among other things, novel methods used to process batches of transformer-based requests, which involves scheduling batches at an iteration-level. For example, claim 10 of the ’775 patent recites the claim language below, including claim language (bolded and italicized) directed to iteration-level scheduling:

> 10. A non-transitory computer-readable storage medium storing computer program instructions executable to perform operations for dynamically executing batches of requests on one or more execution engines running a machine-learning transformer model, the operations comprising:
>
> receiving, by a serving system, one or more requests for execution, the serving system including a scheduler and one or more execution engines each coupled to access a machine-learning transformer model including at least a set of decoders;
> ***scheduling, by the scheduler, a batch of requests including the one or more requests for execution on an 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;
> receiving, by a request processor, a new request from a client device, the new request including a sequence of input tokens;
> ***scheduling, by the scheduler, a second batch of requests additionally including the new request for execution on the execution engine,*** the second batch of requests scheduled responsive to determining that the execution engine has memory available to execute the second batch of requests, wherein in a second set of inputs for the second batch of requests, a length of the sequence of input tokens

> for the new request is different from a length of an input for at least one request other than the new request; and
>
> ***generating, by the execution engine, a second set of output tokens*** by applying the transformer model to the second set of inputs for the second batch.

22. The ’775 Patent is directed to an unconventional, non-routine technique for executing batches of requests on an inference system using a transformer model, including for large-scale generative AI. The Patent-in-Suit explains, *inter alia*, that “at one or more iterations, the inference system can modify the batch being executed on the execution engine by adding new incoming requests to the batch or removing completed requests from the batch.” ’775 patent at 2:62-64.

23. The claimed invention overcame problems in the field, including latency and low throughput, by batching and scheduling requests at an iteration-level (i.e., dynamically batching or continuously batching). *See, e.g.*, ’775 Patent at 3:2-6 (patented invention allows for the “response [to] be provided to the user faster” and allows adding new requests to avoid execution engine “being under-utilized”), 5:30-32 (discussing processing request by batches “to achieve higher processor utilization”), 24:28-31 (“dynamic batching allows the serving system 435 to dynamically adjust batches that are processed on execution engines such that the hardware of the execution engine can be fully utilized.”).

24. To the extent any marking or notice was required by 35 U.S.C. § 287, FriendliAI has complied with the marking requirements of 35 U.S.C. § 287. FriendliAI provides websites, which are accessible to the public without charge and which states that PeriFlow is protected by the ’775 Patent. *See* Ex. 5 (https://friendli.ai/patents); Ex. 6 (https://periflow.ai/patents).

25. FriendliAI permits the public to sign-up for and/or use an embodiment of the patented system (“PeriFlow”) via https://friendli.ai/periflow and https://periflow.ai/periflow. *See* https://friendli.ai/periflow; https://periflow.ai/periflow. On both websites, FriendliAI has marked

the Periflow product by stating at the top of the page that "Our engine technology is protected by patents in the United States and Korea," and including a link entitled "Visit patents." By clicking this link, the user is directed to the websites associating PeriFlow—the patented article—with the '775 Patent. *See* Ex. 5 (https://friendli.ai/patents); Ex. 6 (https://periflow.ai/patents).

DEFENDANT'S INFRINGING ACTIVITIES

The Accused Functionality

26. Hugging Face offers an inference server for Large Language Models ("LLMs") called Text Generation Inference ("TGI").

27. On information and belief, Hugging Face launched Text Generation Inference on or around February 2023.

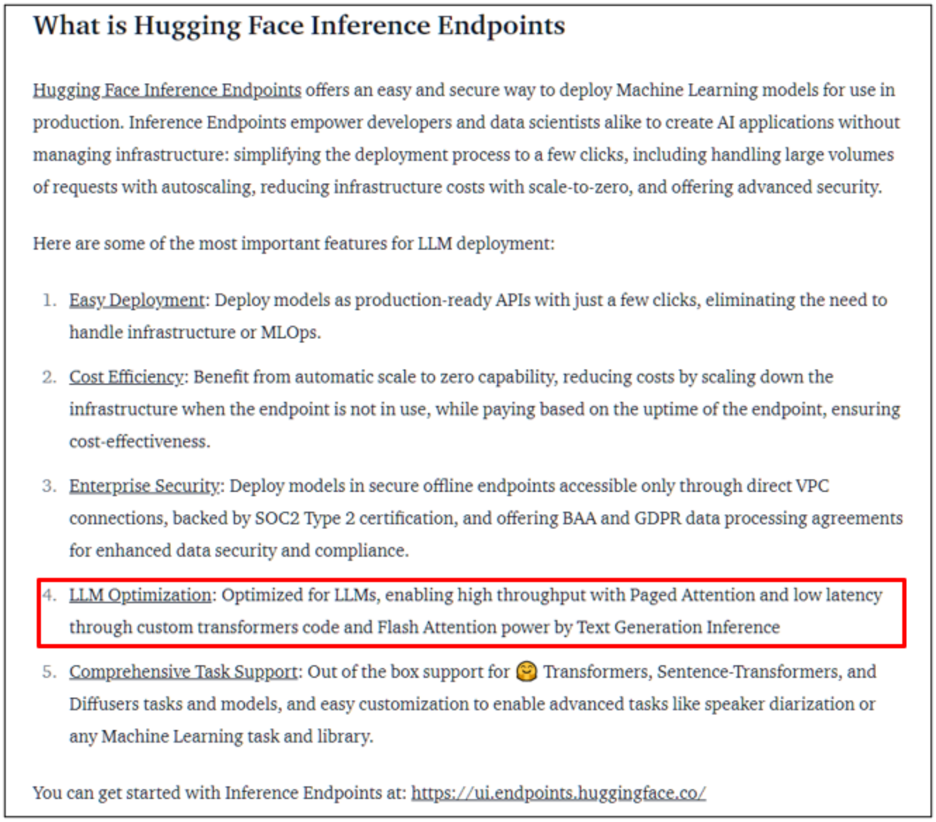
28. An "important feature" of Text Generation Inference is continuous batching of incoming requests. *See* Ex. 7 (https://github.com/huggingface/text-generation-inference/tree/main/router). Hugging Face also refers to continuous batching as "dynamic batching." *See* Ex. 8 (https://huggingface.co/text-generation-inference); *see also* https://github.com/huggingface/text-generation-Noinference/pull/258/commits/c6bb42286e3642261ff9bf7d376687684351c00et (changing dynamic batching's name to continuous batching).

29. Continuous, or dynamic, batching involves "regularly running queries in the same forward step of the LLM (a 'batch') and also removing them when they are finished." *See* Ex. 9.

30. According to Hugging Face, continuous batching enables "increased total throughput." *See* Ex. 10 (https://github.com/huggingface/text-generation-inference#readme). It further allows for optimal balance between "exploiting the hardware and perceived latency," as compared to previously existing inference systems with no continuous batching at all, which suffered from low throughput, and inference systems with static batching, which suffered from

low latency. *See* Ex. 7 (https://github.com/huggingface/text-generation-inference/tree/main/router). *See id.* Inference servers with continuous batching thus allow for obtaining the "sweet spot" of optimal throughput and latency. *See id.*

31. For further example, Defendant's documentation explains how Text Generation Inference, which includes the continuous batching feature, enables high throughput and low latency:



What is Hugging Face Inference Endpoints

Hugging Face Inference Endpoints offers an easy and secure way to deploy Machine Learning models for use in production. Inference Endpoints empower developers and data scientists alike to create AI applications without managing infrastructure: simplifying the deployment process to a few clicks, including handling large volumes of requests with autoscaling, reducing infrastructure costs with scale-to-zero, and offering advanced security.

Here are some of the most important features for LLM deployment:

1. Easy Deployment: Deploy models as production-ready APIs with just a few clicks, eliminating the need to handle infrastructure or MLOps.
2. Cost Efficiency: Benefit from automatic scale to zero capability, reducing costs by scaling down the infrastructure when the endpoint is not in use, while paying based on the uptime of the endpoint, ensuring cost-effectiveness.
3. Enterprise Security: Deploy models in secure offline endpoints accessible only through direct VPC connections, backed by SOC2 Type 2 certification, and offering BAA and GDPR data processing agreements for enhanced data security and compliance.
4. LLM Optimization: Optimized for LLMs, enabling high throughput with Paged Attention and low latency through custom transformers code and Flash Attention power by Text Generation Inference
5. Comprehensive Task Support: Out of the box support for 🤗 Transformers, Sentence-Transformers, and Diffusers tasks and models, and easy customization to enable advanced tasks like speaker diarization or any Machine Learning task and library.

You can get started with Inference Endpoints at: https://ui.endpoints.huggingface.co/

Ex. 11 (https://huggingface.co/blog/inference-endpoints-llm) (emphasis added).

Defendant's Knowledge of the Patent-in-Suit

32. The Patent-in-Suit claims advancements in the large-scale transformer-based AI industry in which Defendant actively participates. The claimed advancements were described in a paper, titled "Orca: A distributed serving system for {Transformer-Based} generative models" (the "Orca paper"), published by Dr. Chun and his colleagues in July 2022. *See* Ex. 3; *see also*

Ex. 12 (https://www.usenix.org/conference/osdi22/presentation/yu). The Orca paper identifies FriendliAI as companies associated with the authors of the Orca paper.

33. FriendliAI's website identifies its inference engine as Periflow, and further identifies Periflow as being also known as Orca. *See* https://www.friendli.ai/; https://medium.com/friendliai/serve-generative-ai-models-like-t5-faster-than-ever-with-orca-32-8x-faster-for-t5-3b-a6ae560cfe25.

34. FriendliAI's press release further identifies the Patent-in-Suit, and identifies that Periflow (aka Orca) practices the Patent-in-Suit. *See* Ex. 13 (https://www.businesswire.com/news/home/20230720505202/en/FriendliAI-Launches-Public-Beta-of-PeriFlow-Cloud).

35. On information and belief, Defendant has been and is aware of the Orca paper and/or the technology described in the Orca paper. On information and belief, Defendant copied the technology described in the Orca paper in developing TGI, including the continuous batching (or dynamic batching) feature.

36. A third-party article describing continuous batching (also known as dynamic batching or iteration-level scheduling) asserts that the Orca paper appears to be the "first" to describe such technique, and further recognizes that Defendant uses such infringing technique in TGI. *See* Ex. 4 (https://www.anyscale.com/blog/continuous-batching-llm-inference). On information and belief, Defendant has been and is aware of the aforementioned third-party article.

37. On or around July 21, 2023, FriendliAI's counsel notified Defendant that Defendant is infringing the '775 patent and specifically identified how Defendant's TGI infringes the '775 patent. FriendliAI's counsel further notified Defendant that FriendliAI's PeriFlow solution (also known as Orca), practices the '775 patent, and provided a copy of the press release explaining that

PeriFlow is protected by the ’775 patent. FriendliAI’s counsel demanded that Defendant cease and desist its infringing acts.

38. On information and belief, Defendant is and has been aware of the Patent-in-Suit and the fact that Defendant’s TGI practices the claimed invention of the ’775 Patent. Despite its knowledge of the Patent-in-Suit and of its infringement of that Patent, and despite its knowledge that FriendliAI’s PeriFlow solution is protected by the Patent-in-Suit, Defendant has continued to willfully infringe the Patent-in-Suit, obtaining the significant benefits of FriendliAI’s innovation without seeking FriendliAI permission or paying any compensation to FriendliAI for access to this valuable technology. For example, Defendant has continued to use TGI, including the continuous batching (or dynamic batching) feature, without a license after receiving the letter from FriendliAI’s counsel, even though that letter notified Defendant of its infringing acts and demanded that Defendant cease and desist.

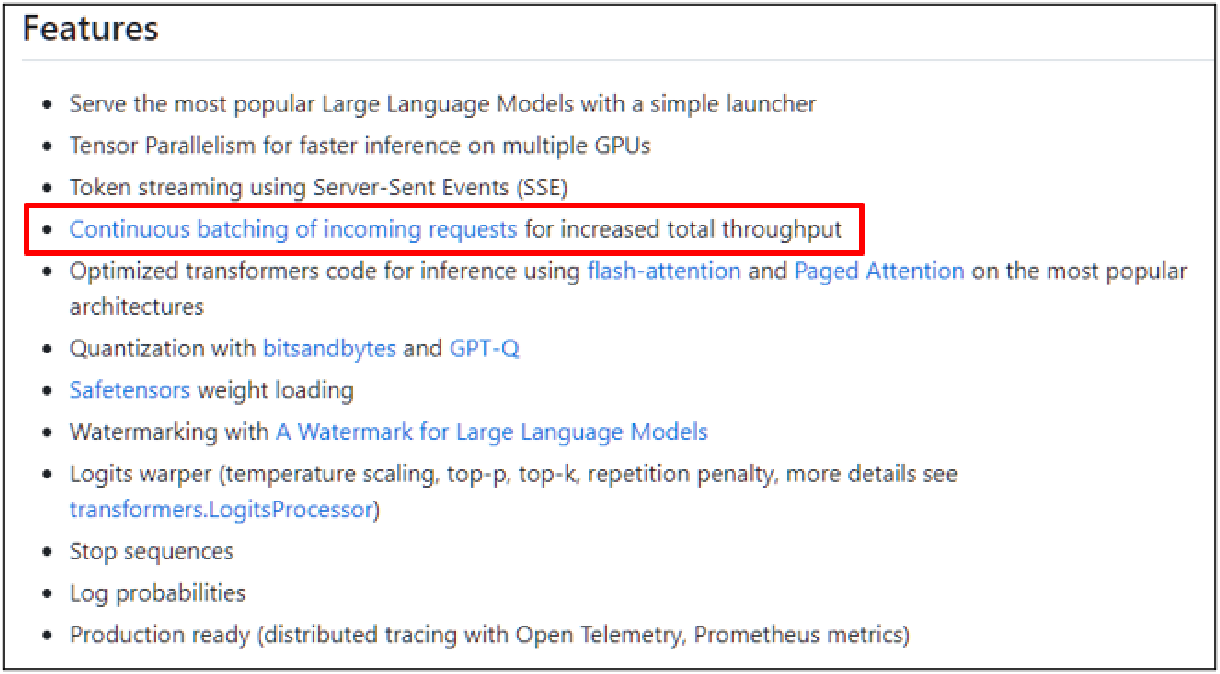
COUNT I: INFRINGEMENT OF THE ’775 PATENT

39. FriendliAI incorporates by reference paragraphs 1-38.

40. The ‘775 Patent is valid and enforceable.

41. Defendant has infringed, and continues to infringe, one or more claims of the ’775 Patent under 35 U.S.C. § 271, either literally and/or under the doctrine of equivalents, by making, using, selling, and/or offering for sale by those claims, including products and services that incorporate Text-Generation Inference (the “Accused Functionality” or “TGI”). Defendant’s products and services that incorporate the Accused Functionality (“the Accused Products and Services”) include, but are not limited to, Spaces, Inference Endpoints, Enterprise Hub (formerly known as Private Hub), HuggingChat, OpenAssistant, and Docker Hub containers.

42. As one example, Defendant infringes one or more claims of the ’775 Patent by using TGI, which includes the feature of continuous batching, or dynamic batching:



Features

- Serve the most popular Large Language Models with a simple launcher
- Tensor Parallelism for faster inference on multiple GPUs
- Token streaming using Server-Sent Events (SSE)
- Continuous batching of incoming requests for increased total throughput
- Optimized transformers code for inference using flash-attention and Paged Attention on the most popular architectures
- Quantization with bitsandbytes and GPT-Q
- Safetensors weight loading
- Watermarking with A Watermark for Large Language Models
- Logits warper (temperature scaling, top-p, top-k, repetition penalty, more details see transformers.LogitsProcessor)
- Stop sequences
- Log probabilities
- Production ready (distributed tracing with Open Telemetry, Prometheus metrics)

Ex. 10 (https://github.com/huggingface/text-generation-inference#features) (emphasis added). Defendant infringes one or more claims of the ’775 Patent through continuous batching or dynamic batching at least by receiving requests for execution on a machine-learning transformer model, scheduling a batch of requests for execution, generating output by applying the transformer model, receiving a new request for execution, executing the new request in a second batch where the new request is a different length than at least one other request, and generating a second set of output tokens. For example, Defendant has infringed, and continues to infringe, representative Claim 10 of the ’775 Patent through continuous batching or dynamic batching, as shown in Exhibit 2.

43. Defendant has infringed, and continues to infringe, one or more claims of the ’775 Patent under 35 U.S.C. § 271(a), either literally and/or under the doctrine of equivalents, by making and/or using the Accused Products and Services. For example, Defendant makes and uses TGI to serve “the most popular Large Language Models” (“LLMs”) within their Spaces, Inference Endpoints, and Enterprise Hub products and services. *See* Ex. 8 (https://huggingface.co/text-

generation-inference). Defendant's documentation explains that TGI is "used in production at HuggingFace to power LLMs api-inference widgets," that are used on each of the Accused Products and Services. Ex. 10 (https://github.com/huggingface/text-generation-inference). For each of the Accused Products and Services, Defendant uses TGI on their own servers to provide services to clients and the public. *See* Ex. 8 (https://huggingface.co/text-generation-inference). For example, Defendant makes and provides a "Chat UI" to the public that uses TGI and is hosted on Spaces, that runs Defendant's servers. *Id.* For further example, Defendant also uses the Accused Functionality in publicly available products and services like HuggingChat and OpenAssistant. *See* https://huggingface.co/blog/sagemaker-huggingface-llm; https://huggingface.co/chat; https://open-assistant.io. For further example, Defendant also uses the Accused Functionality to provide services to customers using the Spaces, Inference Endpoints, and Enterprise Hub products and services. *See, e.g.*, Ex. 8 (https://huggingface.co/text-generation-inference); *see also* https://huggingface.co/spaces; https://huggingface.co/blog/inference-endpoints-llm; https://huggingface.co/docs/inference-endpoints/main/en/others/runtime#inference-endpoints-version; https://huggingface.co/blog/introducing-private-hub. For example, Defendant's documentation indicates that Text Generation Inference on Inference Endpoints is executed on Defendant's servers. *See* https://huggingface.co/blog/inference-endpoints-llm. For further example, Defendant's documentation indicates that Enterprise Hub allows customers to deploy services like Inference Endpoints, which uses TGI and is, on information and belief, hosted by Defendant on Defendant's servers. *See* https://huggingface.co/blog/introducing-private-hub.

44. Defendant has also infringed, and continues to infringe, one or more claims of the '775 Patent under 35 U.S.C. § 271(a), either literally and/or under the doctrine of equivalents, by

selling, offering for sale, and/or otherwise making available the Accused Products incorporating the Accused Functionality. For example, Defendant sells, offers for sale, and otherwise makes available the Accused Products and Services, including but not limited to Spaces, Inference Endpoints, and Enterprise Hub. *See* https://huggingface.co/pricing; https://huggingface.co/docs/inference-endpoints/main/en/others/runtime#inference-endpoints-version; https://huggingface.co/blog/inference-endpoints-llm; https://huggingface.co/blog/introducing-private-hub; https://huggingface.co/spaces. For further example, Defendant also makes available the Docker Hub container for customers to download and use the Accused Functionality on their own system. *See* Ex. 10 (https://github.com/huggingface/text-generation-inference#docker).

45. In addition or in the alternative, Defendant has also induced infringement, and continues to induce infringement, of one or more claims of the ’775 Patent under 35 U.S.C. § 271(b). Defendant actively, knowingly, and intentionally induces infringement of the ’775 Patent by selling or otherwise making available the Accused Products and Services, with the knowledge and intent that third-party customers and users will use the Accused Products and Services, including the Accused Functionality, made available by Defendant to infringe the ’775 Patent. Defendant acts with the knowledge and intent to encourage and facilitate third-party infringement through the dissemination of the Accused Products and Services and/or the creation and dissemination of supporting materials, documentation, instructions, code and/or technical information related to the Accused Products and Services, including the Accused Functionality.

46. Defendant specifically intends and is aware that the ordinary and customary use of the Accused Products and Services would infringe the ’775 Patent. For example, Defendant sells and provides the Accused Products and Services, which when used in their ordinary and customary

manner intended and instructed by Defendant, infringes one or more claims of the ’775 Patent, including at least claim 10. On information and belief, Defendant further provides supporting materials, documentation, instructions, code and/or technical information that cause their customers and partners to operate the Accused Products and Services for their ordinary and customary use. *See, e.g.*, Ex. 8 (https://huggingface.co/text-generation-inference/); *see also* https://huggingface.co/blog/falcon; https://huggingface.co/blog/inference-endpoints-llm; https://github.com/huggingface/text-generation-inference#docker. Defendant accordingly induces third parties to use the Accused Products and Services in their ordinary and customary way to infringe the ’775 Patent, knowing, or at least being willfully blind to the fact, that such use constitutes infringement of the ’775 Patent.

47. In addition or in the alternative, Defendant contributes to the infringement by third parties, such as customers and users, of one or more claims of the ’775 Patent under 35 U.S.C. § 271(c), by making, selling and/or offering for sale in the United States, and/or importing into the United States, Accused Products, which include the Accused Functionality, knowing that those products constitute a material part of the inventions of the ’775 Patent, knowing that those products are especially made or adapted to infringe the ’775 Patent, and knowing that those products are not staple articles of commerce suitable for substantial non-infringing use.

48. On information and belief, Defendant has had knowledge of and notice of the ’775 Patent and its infringement since at least the launch date of the Accused Functionality, and no later than July 21, 2023 when, on information and belief, Defendant received a letter notifying Defendant of the ’775 Patent and its infringement.

49. On information and belief, Defendant’s infringement of the ’775 Patent has been, and continues to be, willful and deliberate since at least the launch date of the Accused

Functionality, and no later than July 21, 2023 when, on information and belief, Defendant received a letter notifying Defendant of the ’775 Patent and its infringement.

50. FriendliAI has been and continues to be damaged by Defendant’s infringement to the ‘775 Patent and will suffer irreparable injury unless the infringement is enjoined by this Court.

51. Defendant’s conduct in infringing the ’775 Patent renders this case exceptional within the meaning of 35 U.S.C. § 285.

PRAYER FOR RELIEF

WHEREFORE, FriendliAI prays for judgment as follows:

A. That Defendant has infringed the Patent-in-Suit.

B. That Defendant’s infringement of the Patent-in-Suit has been willful;

C. That FriendliAI be awarded all damages adequate to compensate it for Defendant’s past infringement and any continuing or future infringement of the Patent-In-Suite up until the date such judgment is entered, including pre- and post-judgment interest, costs, and disbursements as justified under 35 U.S.C. § 284;

D. That any award of damages be enhanced under 35 U.S.C. § 284 as result of Defendant’s willful infringement;

E. That this case be declared an exceptional case within the meaning of 35 U.S.C. § 285 and that FriendliAI be awarded the attorney fees, costs, and expenses incurred in connection with this action;

F. That Hugging Face’s infringing acts be enjoined, or, at least, be subject to a compulsory ongoing licensing fee; and

F. That FriendliAI be awarded such other and further relief at law or equity as this Court deems just and proper.

DEMAND FOR JURY TRIAL

Plaintiff FriendliAI hereby demands a trial by jury on all issues so triable.

OF COUNSEL:

Michael J. Sacksteder
Shreyas A. Kale
John M. DiBaise
FENWICK & WEST LLP
555 California Street, 12th Floor
San Francisco, CA 94104
Tel: (415) 875-2300
MSacksteder@fenwick.com
SKale@fenwick.com
MDiBaise@fenwick.com

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

Dated: July 28, 2023
10940230/

Respectfully,

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.