

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NVIDIA CORPORATION,
Petitioner,

v.

NEURAL AI, LLC,
Patent Owner.

Case IPR2025-00608
Patent RE49,461

PATENT OWNER'S PRELIMINARY RESPONSE

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. PETITIONER FAILS TO DEMONSTRATE OBVIOUSNESS OVER ANY GROUND (ALL CLAIMS, ALL GROUNDS).	3
A. Petitioner Fails To Demonstrate That A POSITA Would Be Motivated Or Able To Combine Buck/Wilt (All Claims, All Grounds).	3
1. Petitioner Never Explains What The Combination Actually Is.	4
2. Petitioner’s Asserted Motivation To Combine Buck and Wilt Is Legally And Factually Insufficient.	8
3. Petitioner Fails To Show That Wilt Would Improve—Or Even Fit Into—Buck’s Task Management System.	12
B. Petitioner Fails To Demonstrate That Numerous Limitations Are Met (All Claims, All Grounds).	17
1. Petitioner Fails To Demonstrate That Numerous “User Interaction Stream” Limitations Are Met (All Claims, All Grounds).	18
a. Petitioner Fails To Clearly Identify What The “User Interaction Stream” Is In Buck/Wilt.	19
b. Petitioner Fails To Demonstrate That Buck/Wilt Discloses Or Renders Obvious A “User Interaction Stream.”	24
c. Petitioner Fails To Demonstrate That Any Alleged “User Interaction Stream” Performs The Claimed Functions.	27

2.	Petitioner Fails To Consistently Identify What In Buck/Wilt Satisfies The “Computational Stream” Limitations (All Claims, All Grounds).	30
a.	Petitioner Fails To Clearly And Consistently Identify A “Computational Stream.”	32
b.	None Of Petitioner’s Options Perform The Required Functions.	35
c.	Petitioner Improperly Equates The “Computational Stream” With The “Controller.”	37
3.	Petitioner Fails To Show “The Computational Stream Controlling Data Exchange” Between The Streams (Claims 21–32, 38–41 All Grounds).	38
a.	Petitioner Fails To Prove That Buck Discloses The Computational Stream Controlling Data Exchange During Execution Of Computations.	39
b.	Petitioner Fails To Show That Wilt Discloses The Limitations.	42
c.	Petitioner’s Preemption Theory Also Fails.	44
4.	Petitioner Fails To Demonstrate That The “User Interaction Stream Controlling Transfer Of The Input Data” To The Memory Partitions “During Execution Of The Computations” (Claims 33–37, All Grounds).	48
5.	Petitioner Fails To Demonstrate That The “User Command” Limitation Is Disclosed And/Or Obvious (All Claims, All Grounds).	50
III.	CONCLUSION	52

TABLE OF AUTHORITIES

Page(s)

COURT DECISIONS

<i>ActiveVideo Networks, Inc. v. Verizon Communs., Inc.</i> , 694 F.3d 1312 (Fed. Cir. 2012)	10
<i>Aspex Eyewear, Inc. v. Marchon Eyewear, Inc.</i> , 672 F.3d 1335 (Fed. Cir. 2012)	38
<i>Bd. Of Regents v. BENQ Am. Corp.</i> , 533 F.3d 1362 (Fed. Cir. 2008)	34, 38
<i>CAE Screepnplates Inc. v. Heinrich Fiedler GmbH & Co.</i> , 224 F.3d 1308 (Fed. Cir. 2000)	34
<i>Corephotonics, Ltd. v. Apple Inc.</i> , 84 F.4th 990 (Fed. Cir. 2023)	23
<i>Harmonic Inc. v. Avid Tech., Inc.</i> , 815 F.3d 1356 (Fed. Cir. 2016)	8
<i>In re Ratti</i> , 270 F.2d 810 (CCPA 1959)	16
<i>In re Schweickert</i> , 676 Fed. App'x. 988 (Fed. Cir. 2017)	17
<i>Intelligent Bio-Sys., Inc. v. Illumina Cambridge, Ltd.</i> , 821 F.3d 1359 (Fed. Cir. 2016)	24
<i>KSR Int'l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007)	11
<i>Microsoft Corp v. Biscotti, Inc.</i> , 878 F.3d 1052 (Fed. Cir. 2017)	23

<i>Netflix, Inc. v. DivX, LLC</i> , 84 F.4th 1371 (Fed. Cir. 2023)	23, 35
<i>Par Pharm., Inc. v. TWi Pharms., Inc.</i> , 773 F.3d 1186 (Fed. Cir. 2014)	25
<i>Plas-Pak Indus. v. Sulzer Mixpac AG</i> , 600 Fed. App'x. 755 (Fed. Cir. 2015)	16
<i>Sisvel Int'l S.A. v. Sierra Wireless, Inc.</i> , 82 F.4th 1355 (Fed. Cir. 2023)	23
<i>Virtek Vision Int'l ULC v. Assembly Guidance Sys., Inc.</i> , 97 F.4th 882 (Fed. Cir. 2024)	12, 16

AGENCY DECISIONS

<i>Hulu, LLC v. Sound View Innovations, LLC</i> , IPR2018-00582, Paper 34 (Aug. 5, 2019) (informative)	10, 17
<i>Markforged Inc., v. Continuous Composites, Inc.</i> , IPR2022-01220, Paper 8 (Jan. 11, 2023).....	8
<i>Microsoft Corp. v. Virtru Corp.</i> , IPR2023-00019, Paper 12, (May 17, 2023).....	8
<i>Samsung Elecs. Co., Ltd. v. Red Rock Analytics, LLC</i> , IPR2018-00556, Paper 18 (Aug. 20, 2018)	10
<i>Xerox Corp. v. Bytemark, Inc.</i> , IPR2022-00624, Paper 12 (Feb. 10, 2023) (Director Review)	22
<i>Xerox Corp. v. Bytemark, Inc.</i> , IPR2022-00624, Paper 9 (Aug. 24, 2022) (precedential)	22

STATUTES

35 U.S.C. § 312(a)(3)..... 23, 35

EXHIBIT LIST	
2001	Lowenstein Declaration In Support of Notice of Intent
2002	Woo Declaration In Support of Notice of Intent
2003	Declaration of Dr. Massimiliano Versace in Support of Patent Owner's Discretionary Denial Brief [Versace-Decl.]
2004	Email from Massimiliano Versace (Monday, June 26, 2017 1:58:15 PM) [Email-Chain]
2005	Neurala_presentation_nVidia_6_2017.pdf (Redacted) [Neurala-Presentation]
2006	First Amended Complaint For Patent Infringement in <i>Neural AI, LLC v. NVIDIA Corp.</i> , No. 7:24-cv-00221 (W.D. Tex.), Dkt. 30, December 12, 2024 [Amended-Complaint]
2007	Defendant's Answer To Plaintiff's Amended Complaint in <i>Neural AI, LLC v. NVIDIA Corp.</i> , No. 7:24-cv-00221 (W.D. Tex.), Dkt. 31, December 23, 2024 [Amended-Answer]
2008	Scheduling Order in <i>Neural AI, LLC v. NVIDIA Corp.</i> , No. 7:24-cv-00221 (W.D. Tex.), Dkt. 39, January 27, 2025 [Scheduling-Order]
2009	Defendant NVIDIA Corporation's Identification of Preliminary Claim Constructions served April 15, 2025 in <i>Neural AI, LLC v. NVIDIA Corp.</i> , No. 7:24-cv-00221 (W.D. Tex.) [NVIDIA-Preliminary-Claim-Constructions]
2010	Defendant's NVIDIA Corporation's Opening Claim Construction Brief in <i>Neural AI, LLC v. NVIDIA Corp.</i> , No. 7:24-cv-00221 (W.D. Tex.), Dkt. 71 [NVIDIA-Claim-Construction-Brief]
2011	Defendant NVIDIA Corporation's Preliminary Invalidity Contentions served April 1, 2025 in <i>Neural AI, LLC v. NVIDIA Corp.</i> , No. 7:24-cv-00221 (W.D. Tex.) (Redacted) [Invalidity-Contentions-Cover]

2012	Ex. C12 – Invalidity of U.S. Patent No. RE49,461 – Buck ’977 served April 1, 2025 in <i>Neural AI, LLC v. NVIDIA Corp.</i> , No. 7:24-cv-00221 (W.D. Tex.) [Invalidity-Contentions-Buck]
2013	Ex. C9 – Invalidity of U.S. Patent No. RE49,461 – Wilt served April 1, 2025 in <i>Neural AI, LLC v. NVIDIA Corp.</i> , No. 7:24-cv-00221 (W.D. Tex.) [Invalidity-Contentions-Wilt]
2014	Ex. C11 – Invalidity of U.S. Patent No. RE49,461 – GPU Gems served April 1, 2025 in <i>Neural AI, LLC v. NVIDIA Corp.</i> , No. 7:24-cv-00221 (W.D. Tex.) [Invalidity-Contentions-GPU-Gems]
2015	Exhibit C13 – Invalidity of U.S. Patent No. RE49,461 – nnet served April 1, 2025 in <i>Neural AI, LLC v. NVIDIA Corp.</i> , No. 7:24-cv-00221 (W.D. Tex.) [Invalidity-Contentions-nnet]
2016	Ex. C14 – Invalidity of U.S. Patent No. RE49,461 – ANN served April 1, 2025 in <i>Neural AI, LLC v. NVIDIA Corp.</i> , No. 7:24-cv-00221 (W.D. Tex.) [Invalidity-Contentions-ANN]
2017	U.S. Patent No. 8,922,566 [NVIDIA-’566-patent]
2018	Excerpt of File History for U.S. Patent No. 8,922,566 [NVIDIA-’566-FH]
2019	Excerpt of File History for U.S. Patent Application No. 13/335,850 [13/335,850-FH]
2020	The Real Brains Behind the Artificial Intelligence, NEURALA, https://www.neurala.com/about-may (last visited June 14, 2025) [Neurala-About]
2021	Nefi Alarcon, “Inception Spotlight: AI Startup Neurala Sees 7X Speedup with NGC,” NVIDIA DEVELOPER (Sept. 25, 2019), https://developer.nvidia.com/blog/inception-spotlight-ai-startup-neurala-sees-7x-speedup-with-ngc/ [NVIDIA-Blog-Post]

2022	Murali Gopalakrishna, LINKEDIN, https://www.linkedin.com/in/muralimohang/ (last visited June 12, 2025) [Gopalakrishna-LinkedIn]
2023	Kashi Anoosheh, LINKEDIN, https://www.linkedin.com/in/kashianoosheh/ (last visited June 12, 2025) [Anoosheh-LinkedIn]
2024	Jeff Herbst, GFT VENTURES, https://www.gft.vc/jeff-herbst (last visited June 14, 2025) [Herbst-Bio]
2025	Standing Order Governing Proceedings (OGP) 4.4—Patent Cases, W.D. Tex (Jan. 23, 2024) [Standing-Order-Governing-Proceedings]
2026	Screenshot of Time To Trial – Alan D. Albright, DOCKETNAVIGATOR (last visited June 13, 2025) [Albright-Time-To-Trial]
2027	U.S. Patent No. 7,219,085 [Buck-085]
2028	Central Reexamination Unit, USPTO, https://www.uspto.gov/about-us/organizational-offices/office-commissioner-patents/central-reexamination-unit (last visited June 8, 2025) [CRU]
2029	U.S. Patent No. 8,648,867 [’867-Patent]
2030	Declaration of Sunil Khatri, Ph.D.

I. INTRODUCTION

The Petition should be denied as it fails to demonstrate a reasonable likelihood of prevailing on any challenged claim. Indeed, it fails for multiple, independently sufficient reasons.

The claims are directed to systems in which two distinct execution streams—a “user interaction stream” executed by a CPU and a “computational stream” executed by a processing unit—manage artificial neural network computation and user command execution. Each asserted ground rests on a proposed combination of Buck and Wilt (“Buck/Wilt”). Pet., 3. Critically, however, Petitioner fails to articulate a coherent or legally supportable combination of Buck and Wilt. As shown in Section II.A, the Petition does not explain how Buck’s system “for processing and execution of general matrix operations” (Ex. 1004 [Buck] [0002]) would be modified by, or even interact with, Wilt’s teachings of “managing the computational resources of coprocessor(s), such as [GPUs]” (Ex. 1005 [Wilt] Abstract). Instead, Petitioner offers only vague and conclusory statements, with no explanation of what the resulting system would be or how it would operate. *See* Section II.A.1. The Petition also fails to provide a reasoned motivation to combine: it identifies no deficiency in Buck, no specific benefit from Wilt, and no reason a POSITA would combine the two. *See* Section II.A.2. Worse still, the combination would affirmatively undermine Buck’s system. Buck is

expressly designed to minimize memory transfers between CPU and GPU through a multi-pass structure that keeps data localized within GPU memory. Wilt’s virtualization model, by contrast, relies on those very transfers, and manages the computational resources in such a scenario. The proposed combination is not only undefined—the identified elements of Buck and Wilt are incompatible. *See* Section II.A.3.

Petitioner’s failure to present a coherent combination causes even more problems for Petitioner. Petitioner does not clearly identify what in Buck, Wilt, or the combination corresponds to either the “user interaction stream” or the “computational stream.” Nor does it show that anything in Buck or Wilt performs the functions required of those streams, or that the claimed data exchanges between them even occur. *See* Sections II.B.1–4. Petitioner also fails to demonstrate that the “user command” limitations are disclosed or rendered obvious. *See* Section II.B.5.

These failures—the failure to articulate a viable or supported combination and the inability to show how the prior art maps to the claims—are independently fatal. The Petition never reaches the threshold showing of a reasonable likelihood of success on any claim. For each of these reasons, institution should be denied.

II. PETITIONER FAILS TO DEMONSTRATE OBVIOUSNESS OVER ANY GROUND (ALL CLAIMS, ALL GROUNDS).

Petitioner's grounds should be rejected for multiple reasons. Petitioner fails to specify its proposed combination or adequately explain why a POSITA would be motivated to make it. *See* Section II.A. Also, whereas the claims focus on the interplay between a “user interaction stream” and a “computational stream,” Petitioner fails to demonstrate that its Buck/Wilt combinations disclose or render obvious either stream, or numerous other limitations as well. *See* Section II.B.

A. Petitioner Fails To Demonstrate That A POSITA Would Be Motivated Or Able To Combine Buck/Wilt (All Claims, All Grounds).

Fundamentally, the Petition fails to clearly present the proposed Buck/Wilt combination which forms the basis of all of Petitioner's grounds. Institution should be denied for three independently sufficient reasons.

First, Petitioner never specifies what its combination actually is, offering only vague descriptions without explaining how Buck's system for “processing and execution of general matrix operations” and used for GPU-accelerated neural networks would be modified by, or even relate to, Wilt's multi-application resource scheduler. *See* Section II.A.1. Second, the ill-defined combination relies only on generic, conclusory motivations that are neither evidenced in the prior art nor tethered to any supposed modification. *See* Section II.A.2. Third, the

combination rests on the false premise, that Buck is silent on GPU task management, when in fact, Buck’s “multi-pass” architecture and focus on “execut[ing] general matrix operations” (*see* Ex. 1004 [Buck] [0010], [0155]) is expressly designed to manage GPU operations locally and to avoid costly memory transfers between the CPU and the GPU—precisely the type of transfers on which Wilt’s methods rely. *See* Section II.A.3. Any one of these deficiencies is fatal to institution.

1. Petitioner Never Explains What The Combination Actually Is.

Each of Petitioner’s asserted grounds depends on a proposed combination of Buck and Wilt. But Petitioner never explains what that combination is. Instead, Petitioner offers several slightly varied statements that all reduce to the same high-level idea: that a POSITA would “implement Wilt’s scheduling and task management system in Buck’s GPU-accelerated system.” *See, e.g.,* Pet., 21; *see also id.*, 22 (“implement Wilt’s resource management method within Buck’s co-processor system”); *id.* (“Wilt supplies the behind-the-scenes details of managing the computational resources of the CPU and the GPU”); *id.*, 24 (combination is “an implementation of a known technique (*e.g.*, Wilt’s resource and task management) within a known system (*e.g.*, Buck’s optimized GPU-accelerated system)”). None of these formulations describe what the proposed system would actually do, how it would operate differently from Buck, what specific deficiencies exist in Buck,

what specific elements from Wilt would be incorporated into which aspects of Buck, or why a POSITA would have been motivated to make such unspecified modifications.

This lack of specificity persists throughout Petitioner's analysis. For example, Petitioner quotes Wilt's discussion of increasing demand for GPU resources and the alleged resulting need to "efficiently manage the complex computing tasks delegated to Buck's GPU." Pet., 23. But it never explains what such "management" would entail in the context of Buck's architecture or what the tasks "delegated to Buck's GPU" and that Buck is unable to handle without such "management" assistance from Wilt are. Notably, Buck does not describe a multitasking environment; it presents a single-task neural network framework which uses "a graphics processing unit (GPU) to execute general matrix operations" structured into multiple "passes" to maximize memory locality and efficiency. *See* Ex. 1004 [Buck] [0010], [0144]–[0155], [0233]–[0244]. Buck's system and method "for optimizing performance of a graphics processing unit (GPU) to execute general matrix operations" already seeks to exploit the parallelism and high memory bandwidth to increase processing efficiency and speed. *See id.*, [0010]. Petitioner never explains whether it proposes to modify Buck's multi-pass structure, change its matrix operations, or do something else entirely. Nor does Petitioner explain what specific aspects of Buck's disclosure

(which does a single task of “execut[ing] general matrix operations”) need improvement and exactly how Wilt’s “scheduling and task management system” would benefit Buck.

As Dr. Khatri explains, the disconnect between Buck (focused on a single application or task like neural network training) and Wilt (focused on resource management of coprocessors in a multitasking environment) further obscures what Petitioner’s combination actually entails:

A POSITA would understand Buck’s focus to be “execut[ing] general matrix operations” (*see* Ex. 1004 [Buck] [0010]) in the context of an application such as neural network training executed on the GPU (*id.*, [0233]–[0244]). More specifically, for such an application, Buck is directed towards avoiding costly memory transfers between the CPU and the GPU by “mak[ing] maximum use of the memory” (*id.*, [0009]) and careful arrangement of “operands and results in graphics memory” into a “sequence of passes” such that “each pass fits into fixed sized memory” (*id.*, [0012]). This is precisely the type of application and careful memory planning that is custom-crafted in a manner that does not require the “management of computational resources ... in a multitasking environment” that is offered by Wilt. *See* Ex. 1005 [Wilt] Abstract.

Similarly, Buck’s “multi-pass” architecture (*see, e.g.*, Ex. 1004 [Buck] [0144]–[0155]) reflects a strategy for optimizing computations within a single task, not coordinating GPU access across multiple tasks.

Petitioner never explains how Wilt’s management of computational resources in a multitasking environment on a shared GPU relates to Buck’s multi-pass execution model, or why a POSITA would seek to combine them.

Ex. 2030 [Khatri-Decl.] ¶¶ 38–39.

Further obscuring the nature of the combination is the lack of any explanation of how Buck would or could be improved by Wilt as Petitioner implies it will be. As Dr. Khatri explains:

For example, it is unclear how Buck’s multi-pass solution of the specific application of matrix operations would be improved—let alone made “more efficient”—by running it in parallel with unrelated applications, which Wilt focuses on. And to the extent that Petitioner means to equate Buck’s “passes” with Wilt’s “applications,” such that Wilt’s scheduler would somehow apply across Buck’s passes, that analogy is neither articulated by Petitioner nor technically sound. ...

Ex. 2030 [Khatri-Decl.] ¶ 40.

Petitioner’s limitation-by-limitation analysis offers no further clarity. For example, in addressing limitation 21[b], Petitioner begins with the statement that “Buck/Wilt discloses this limitation.” Pet., 38. This is followed by two paragraphs discussing Buck (*id.*, 39–40) and then three paragraphs discussing Wilt (*id.*, 40–42). In the Wilt discussion, Buck is not mentioned, except in a cryptic reference to “the Buck/Wilt system” as supposedly explained by Petitioner’s analysis of the

preamble (21[pre]) which does not mention Wilt or explain what exactly from each reference is combined (*id.*, 31). *Id.*, 40–41.

This siloed presentation consisting of separate discussions of Buck and Wilt with no attempt to explain how their teachings are combined, leaves the supposed “Buck/Wilt” system undefined. Ex. 2030 [Khatri-Decl.] ¶¶ 42–43. And, as a matter of law, disconnected discussions of different references are insufficient to establish obviousness. *Microsoft Corp. v. Virtru Corp.*, IPR2023-00019, Paper 12, 18 (May 17, 2023) (“obviousness cannot be shown merely by pointing to disparate features of a disclosure.”) (denying institution); *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016) (“it was [Petitioner’s] burden to explain to the Board how Haskell combined with Rossmere rendered the challenged claims unpatentable.”); *Markforged Inc. v. Continuous Composites, Inc.*, IPR2022-01220, Paper 8, 36 (Jan. 11, 2023) (similar) (denying institution).

Because Petitioner fails to identify any coherent integration of Buck and Wilt, it has not demonstrated a reasonable likelihood of prevailing under any ground.

2. Petitioner’s Asserted Motivation To Combine Buck and Wilt Is Legally And Factually Insufficient.

As explained above, Petitioner never articulates what its proposed combination of Buck and Wilt actually is. It never identifies what in Buck would

be changed, how Wilt would be applied, or how the resulting system would function. Against this undefined backdrop, Petitioner's asserted motivations to combine are not just conclusory and unsupported by express teaching or disclosure in the prior art, they are untethered to any specific architectural proposal. Without explaining what the combination *is*, Petitioner cannot meaningfully explain why a POSITA would have been motivated to make it. Indeed, the prior art simply lacks any motivation to combine or modify these different prior art systems of Buck and Wilt; Petitioner's conclusory attorney argument and Dr. Rosing's parroting declaration cannot fix this fatal flaw.

Petitioner offers several high-level and unsupported reasons why a POSITA supposedly would have combined Buck and Wilt: that Buck lacks scheduling detail and Wilt supplies it¹ (Pet., 21–22); that both references come from Microsoft and use similar hardware architectures (*id.*, 22); that features are sometimes spread across different patent applications (*id.*); that the combination would improve efficiency (*id.*, 23–24); and that Wilt reflects a “known technique” yielding “predictable results” (*id.*, 24).

Petitioner, however, never connects these generalized rationales to a specific shortcoming in Buck or a defined technical gain from Wilt. Nor are these generic

¹ And, as further discussed in Section II.A.3, this is not, in fact, true.

rationales evidenced by the prior art. Petitioner never shows or even claims that Buck's multi-pass execution approach for a specific mathematical task (matrix operations) is inadequate, inefficient, or outdated. Nor does it show that Wilt's resource and task scheduler for multitasking environments was ever applied to or suitable for the single-task context presented in Buck. Thus, a POSITA would not have had any reason to modify either reference in view of the other, let alone combine them in a way Petitioner fails to specify.

Asserting that a system could be “optimized” (Pet., 21–22) or result in “efficient[] [resource] shar[ing]” (*id.*, 22–24) is not a substitute for a reasoned basis to alter a specific architecture. *See, e.g., ActiveVideo Networks, Inc. v. Verizon Communs., Inc.*, 694 F.3d 1312, 1327–28 (Fed. Cir. 2012) (generic motivation to “build something better” “more efficient, cheaper,” etc. was “insufficient”); *Samsung Elecs. Co., Ltd. v. Red Rock Analytics, LLC*, IPR2018-00556, Paper 18, 22 (Aug. 20, 2018) (“statements of increased utility and minimal modifications are generic, and fail to provide necessary factual support—they are akin to stating in a conclusory fashion that the combination ‘would have been obvious.’”); *Hulu, LLC v. Sound View Innovations, LLC*, IPR2018-00582, Paper 34, 21–22 (Aug. 5, 2019) (informative) (“Petitioner’s reliance on a generic desire for ‘improved access times’ in the Reply strikes us as merely a bald statement about what could have been achieved at the time of the invention.”).

Petitioner also implies that shared Microsoft authorship supports the combination. Pet., 22. But that inference has no evidentiary basis. Petitioner does not identify any shared inventors, cross-citations, or claimed relationship between Buck and Wilt. The references do not describe each other or suggest that they were developed together. Absent any such linkage, the fact that both were filed by the same multinational assignee is irrelevant to a POSITA's motivation to combine them. Indeed, when each reference is sufficiently analyzed, as above, it is clear that these references are not combinable.

Petitioner's invocation of *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398 (2007), likewise falls short. Petitioner characterizes the proposed combination as “an implementation of a known technique ... within a known system” that would “yield predictable results.” Pet., 24. But *KSR* makes clear that such reasoning applies only when the record shows contextual grounding: “design incentives,” “market forces,” or a well-established practice of applying the technique in analogous systems. *See id.*, 417–18. In *KSR*, the patent challenger showed that the prior art system had a recognized limitation, that known sensor integration techniques were widely applied in the field, and that the market demanded such an upgrade. *See id.*, 424. Petitioner offers no such factual basis here—just the bare assertion that scheduling is good. *E.g.*, Pet., 21 (“A POSITA would have been motivated to implement Wilt's scheduling and task management system in Buck's

GPU-accelerated system for simulating artificial neural networks *to optimize the performance of the system.*”).

The Federal Circuit has repeatedly rejected this kind of logic. In *Virtek Vision Int’l ULC v. Assembly Guidance Sys., Inc.*, 97 F.4th 882 (Fed. Cir. 2024), the court reversed an obviousness finding where the petitioner claimed that both coordinate system types were “known in the art,” but failed to explain why a POSITA would choose to substitute one for the other. *Id.*, 886–87. Here too, Petitioner asserts that Wilt’s scheduler was “known,” but never explains why a POSITA would inject that complexity into Buck’s purpose-built, single-application GPU execution model.

In short, Petitioner fails to specifically explain what Buck lacks, what Wilt fixes, or how the claimed benefits would be achieved. A POSITA might “look[] to other art,” as Petitioner claims (Pet., 22), but that is not a reason to combine them, let alone evidence that a POSITA would have done so.

3. Petitioner Fails To Show That Wilt Would Improve—Or Even Fit Into—Buck’s Task Management System.

As discussed above, Petitioner’s Buck/Wilt combination should be rejected because it is ill-defined and unsupported by any credible motivation to combine. But even setting those flaws aside, the combination should still fail because it would disrupt, not improve, Buck’s existing architecture.

Buck addresses the problem of structuring large, complex matrix operation-based computations to minimize data transfers between the host CPU and the GPU and to maximize locality within the GPU. *See, e.g.*, Ex. 1004 [Buck] [0010], [0012]. As Buck explains:

The characteristics of the class of problems that the invention disclosed herein seeks to solve include 1) large matrices or vectors, 2) operations too complex to fit in a single pass, and 3) iteration. In such applications, *there is excessive cost for moving data to or from the host CPU. Thus, it is desirable to do as much computation as possible within the GPU and its memory.*

Ex. 1004 [Buck] [0157].²

To avoid costly memory transfers and maintain as much computation within the GPU as possible, Buck introduces a “multi-pass” structure for a specific single mathematical task of matrix operations in which complex computations are divided into passes, each operating within GPU memory limits, with each pass producing intermediate results consumed by the next. *See id.*, Abstract; cls. 12, 15; [0012], [0144]–[0155]; Ex. 2030 [Khatri-Decl.] ¶ 49. This architecture is designed specifically for a single task (matrix operations) to avoid moving intermediate results out of GPU memory when those results are needed again shortly thereafter.

² All emphases added except where otherwise noted.

As Dr. Khatri explains, Buck’s multi-pass structure is not merely a programming convenience. It is directed towards addressing the inefficiencies of memory transfers between the GPU and CPU. Ex. 2030 [Khatri-Decl.] ¶ 50. Buck does not ignore GPU scheduling; it addresses it directly, by explicitly embedding it in the multi-pass algorithm structure, thus ensuring that data is kept in place and processed over a series of passes, yielding efficient use of memory and “greater efficiency and speed in processing and executing general matrix operations.” *See* Ex. 1004 [Buck] [0012].

While Buck seeks to avoid memory transfers between the CPU and GPU, such transfers are precisely how Wilt manages GPU resources across “multiple applications in a multitasking environment.” *See* Ex. 1005 [Wilt] Abstract. Wilt describes a system that virtualizes GPU resources, such as video memory surfaces and buffers, allowing applications to operate as if more memory is available than physically exists. Wilt teaches that physical memory that is not needed is “paged” out to disk, only paging in video memory surfaces and buffers as needed. Wilt teaches that “[i]n the context of this invention, GPU resources such as video memory” can be managed analogous to virtual memory. *Id.*, [0142]–[0143]; Ex. 2030 [Khatri-Decl.] ¶ 51.

For example, Wilt explains that when a command buffer requires a surface that has been “paged out” to system memory, the OS must allocate a new surface

in GPU memory, copy the contents from system memory, and “fix up” all references in the command buffer to identify the new location. Ex. 1005 [Wilt][0143]. These memory operations—copying and relinking—are central to Wilt’s method. Ex. 2030 [Khatri-Decl.] ¶ 52.

Buck, by contrast, explicitly warns against such overhead. It states that “transferring intermediate results out of the graphics memory and back to the CPU memory is quite expensive,” and that “dataflow from GPU to CPU is not efficient.” Ex. 1004 [Buck]]0157]–[0158], [0239]. Buck’s multi-pass structure is designed precisely to avoid these transfers by keeping intermediate results in GPU memory, using known layouts and physical constraints. *See id.*, Abstract, [0012], [0108], [0157]–[0158]; Ex. 2030 [Khatri-Decl.] ¶ 53.

Through its teaching of virtual memory and paging, Wilt teaches resource and task management that is essential to a multi-tasking environment. Buck, by contrast, teaches the optimization of memory for a single specific mathematical task (matrix operations). Ex. 1004 [Buck] [0002], [0012].

A POSITA reading Buck would not view Wilt’s virtualization as an enhancement, but as an unnecessary layer of complexity that interferes with Buck’s direct control of GPU memory without solving any identified problem. *See* Ex. 2030 [Khatri-Decl.] ¶ 54. Buck’s multi-pass structure is expressly designed around physical GPU memory constraints, not abstracted or virtualized memory

resources. Ex. 1004 [Buck] [0012], [0108]; *see also, e.g.*, Ex. 2030 [Khatri-Decl.] ¶ 54.

At best, Petitioner’s combination rests on a false premise that “Buck does not focus on how its system schedules and manages tasks on the GPU, [while] Wilt does.” Pet., 21–22. But, as shown above, this is simply incorrect. Buck schedules and manages GPU execution through its multi-pass model by keeping data within the GPU and avoiding costly transfers between the CPU and GPU. Ex. 2030 [Khatri-Decl.] ¶ 54.

A POSITA, consequently, would not be motivated to change Buck’s “principle of operation” from one that avoids such costly transfers to one that is built upon them. *Plas-Pak Indus. v. Sulzer Mixpac AG*, 600 Fed. App’x. 755, 759 (Fed. Cir. 2015) (“a change in a reference’s ‘principle of operation’ is unlikely to motivate a person of ordinary skill to pursue a combination with that reference”); *In re Ratti*, 270 F.2d 810, 813 (CCPA 1959) (similar).

If Petitioner believes that Buck’s architecture would benefit from Wilt’s virtualization system, it says neither how nor why. It does not identify any deficiency in Buck, any functional gain from Wilt, or any reason a POSITA would expect the result to be beneficial. This is insufficient to establish a motivation. *See Virtek Vision*, 97 F.4th at 887 (“[T]here must exist a motivation to combine various prior art references in order for a skilled artisan to make the claimed invention.”);

Hulu, IPR2018-00582, Paper 34 at 20–21 (rejecting combination where not clear why it would be a “good idea” to modify); *In re Schweickert*, 676 Fed. App’x. 988, 995 (Fed. Cir. 2017) (non-precedential) (remanding where unexplained how “Birrell would have ‘benefitted from the advantages of...Cunniff’s semaphore mechanism.’”).

Thus, for this additional reason, Petitioner fails to demonstrate a POSITA would be motivated to combine Buck and Wilt.

B. Petitioner Fails To Demonstrate That Numerous Limitations Are Met (All Claims, All Grounds).

Petitioner’s failure to clearly lay out its combination also results in Petitioner’s failure to show that numerous limitations are met. The challenged claims recite two distinct execution streams: a “user interaction stream” executed by a CPU and a “computational stream” executed by a processing unit (*e.g.*, a GPU). *See* Ex. 1001 [’461] cls. 21, 33. The “user interaction stream” “control[s] transfer of [] input data” to memory; the “computational stream” “control[s] transfer of the input data from the” memory “during execution of the computations representing the artificial neural network.” *See id.*, 21[a]–[b], 33[d]–[e]. Both claims also require that “user command[s]” or “interaction[s]” received “during execution of the computations representing the artificial neural network” be

executed at times determined by the “computational stream” or “controller”—not immediately. *See id.*, 21[e]–[f], 33[f].

As explained below, Petitioner fails to identify, in a clear and consistent way, what in the prior art constitutes either claimed stream—let alone how they perform the required functions as claimed. *See* Section II.B.1–2. Petitioner also fails to demonstrate that the required data exchanges between the two streams are satisfied and that the user command limitation is disclosed and/or obvious. *See* Section II.B.3–5.

1. Petitioner Fails To Demonstrate That Numerous “User Interaction Stream” Limitations Are Met (All Claims, All Grounds).

Petitioner fails to demonstrate that the prior art discloses or renders obvious the “user interaction stream,” a critical limitation of every challenged claim. Its theory fails for three independent reasons. First, Petitioner fails to articulate a clear or consistent identification of what in Buck, Wilt, or the combination is supposed to be the “user interaction stream.” Second, to the extent Petitioner’s argument can be discerned, it does not demonstrate that the disclosures it points to qualify as a “user interaction stream.” Third, even assuming such a stream were present, Petitioner does not show how the roles the claims require—such as controlling input transfer, coordinating with the computational stream, or executing user commands at times determined by the computational stream—are performed. As

shown below, each of these failures is independently dispositive and each confirm Petitioner's failure to demonstrate a reasonable likelihood of prevailing under any challenged claim.

a. *Petitioner Fails To Clearly Identify What The "User Interaction Stream" Is In Buck/Wilt.*

The claims repeatedly recite a "user interaction stream" executed by the CPU that performs defined roles in the operation of a neural network system, including "controlling transfer of inputs" to memory partitions (claims 21[a], 33[d]), "receiv[ing]" "a user command" or "interaction" (claims 21[e], 33[f]), and coordinating with a "computational stream." Key limitations invoking the "user interaction stream" are highlighted below:

21. A method of executing computations representing an artificial neural network on a computer system comprising at least one central processing unit (CPU), a processing unit, a first memory partition, and a second memory partition, the method comprising: executing, by the at least one CPU, a user interaction stream, the user interaction stream controlling transfer of inputs to the artificial neural network to the first memory partition and the second memory partition; executing, by the processing unit, a computational stream, the computational stream controlling data exchange between the user interaction stream and the computational stream during execution of the computations representing the artificial neural network; shifting control of a data exchange between the user interaction stream and the computational stream to the computational stream in response to starting execution of the computations representing the artificial neural network; shifting control of the data exchange between the user interaction stream and the computational stream to the user interaction stream in response to completion or interruption of the computations representing the artificial neural network; queueing a user command received by the user interaction stream during execution of the computations representing the artificial neural network; and executing the user command during execution of the computations representing the artificial neural network at times determined by the computational stream.	33. A system for executing computations representing an artificial neural network, the system comprising: a camera to acquire input data for the artificial neural network; a first memory partition; a second memory partition; at least one central processing unit (CPU), operably coupled to the camera, the first memory partition, and the second memory partition, to execute a user interaction stream, the user interaction stream controlling transfer of the input data acquired by the camera to the first memory partition and the second memory partition during execution of the computations representing the artificial neural network; a processing unit, operably coupled to the first memory partition, the second memory partition, and the at least one CPU, to execute a computational stream, the computational stream controlling transfer of the input data from the first memory partition and the second memory partition during execution of the computations representing the artificial neural network, the execution of the computations representing the artificial neural network occurring while the camera is acquiring the input data; and a controller, operably coupled to the at least one CPU and the processing unit, to queue user interactions received by the user interaction stream during the execution of the computations representing the artificial neural network for performance at times selected to avoid data corruption.
--	---

Despite this centrality, Petitioner never clearly identifies what in Buck, Wilt, or the combination constitutes the claimed “user interaction stream.”

The Petition begins its discussion of the user interaction stream limitation by pointing to Buck’s generic disclosure that a “user ‘may enter commands and information’ into computer 110 ‘through input devices such as a keyboard 162, and pointing device 161,’” among others. Pet., 34 (citing Ex. 1004 [Buck], [0049]–[0050]). Petitioner then asserts, without elaboration, that “[Buck’s] CPU processes these user interactions and commands by executing a user interaction stream.” *Id.* (citing Ex. 1003 [Rosing-Decl.] ¶ 167). But Buck does not mention any “user

interaction stream,” and Petitioner cites no teaching or element in Buck corresponding to the “user interaction stream.” Petitioner offers no explanation of how that stream is implemented and provides no rationale as to why the generic receipt of user input data through *hardware* devices like a keyboard or pointing device satisfies the claim’s requirement for a *software* “stream” which is executed by the CPU that exercises specific control functions, such as, in claim 21 and similarly in claim 33, “controlling transfer of inputs to the artificial neural network to the first memory partition and the second memory partition.”

Petitioner next cites Buck’s paragraph [0008] as allegedly disclosing that the CPU “maintains its own ‘stream[s]’ of instructions and data.” Pet., 34. But Buck [0008] refers only to the host “CPU’s instruction stream,” not a “user interaction stream.” *See* Ex. 1004 [Buck] [0008] (“The integral floating point co-processor shares the host *CPU’s instruction stream* and has access to CPU memory.”). Nothing in Buck [0008] identifies or suggests a “user interaction stream.” To the extent that Petitioner or Dr. Rosing implicitly maintain that a “user interaction stream” is synonymous with a “host CPU’s instruction stream,” no basis is given for this unstated assumption. And, to the extent that Petitioner contends the CPU instruction stream would carry or have anything to do with the information required for the “user interaction stream,” this too is not shown. *See* Ex. 2030 [Khatri-Decl.] ¶ 63.

Dr. Rosing’s declaration adds nothing of substance. Petitioner cites paragraphs 167 and 168 of her declaration but the former is a lightly edited version of the last paragraph on page 34 of the Petition, conclusorily claiming the presence of a “user interaction stream” without describing what part of Buck maps to this alleged “user interaction stream.” Ex. 1003 [Rosing-Decl.] ¶ 167. Paragraph 168 similarly restates the Petition’s language and notes that Buck’s CPU stores “commands and information” from user interactions through input interface 160, without providing any mapping for the “user interaction stream.” *Id.*, ¶ 168. A parroting declaration such as this, however, which does not identify the bases for its conclusions, is entitled to little weight. *Xerox Corp. v. Bytemark, Inc.*, IPR2022-00624, Paper 9, 15–17 (Aug. 24, 2022) (precedential) (denying institution when expert declaration copies the petition “verbatim” and “adds little to the conclusory assertion for which it is offered to support”); *id.*, Paper 12, 5 (Feb. 10, 2023) (Director Review) (“I determine that the Board was correct in giving little weight to Petitioner’s expert because the expert declaration merely offered conclusory assertions without underlying factual support and repeated, *verbatim*, Petitioner’s conclusory arguments.”) (emphasis in original, citations omitted).

Fundamentally, like the Petition, Dr. Rosing does not identify anything in Buck or Wilt that qualifies as a “user interaction stream,” nor does she attempt to

explain how any aspect of Buck’s system satisfies this limitation. She too does not explain why a “user interaction stream” is the same thing as a “CPU instruction stream.” Ultimately, like the Petition, her testimony simply repeats the conclusory assertion that such a stream is present.

Petitioner’s failure to identify the “user interaction stream” violates its obligation to “identify with particularity” the evidence supporting each challenged claim limitation (35 U.S.C. § 312(a)(3)) and its “burden to present a clear argument.” *Netflix, Inc. v. DivX, LLC*, 84 F.4th 1371, 1377 (Fed. Cir. 2023). Presenting “a clear argument” entails “provid[ing] an understandable explanation of the element-by-element specifics of the patentability challenges, including the identification of particular portions of prior art on which the petitioner is relying.” *See Corephotonics, Ltd. v. Apple Inc.*, 84 F.4th 990, 1001 (Fed. Cir. 2023); *see also Netflix*, 84 F.4th at 1377 (“the Board should [] not have to decode a petition to locate additional arguments beyond the ones clearly made. A petitioner may not rely on a vague, generic, and/or meandering petition and later fault the Board for failing to understand what the petition really meant.”); *Microsoft Corp v. Biscotti, Inc.*, 878 F.3d 1052, 1074 (Fed. Cir. 2017) (petition’s “bare-bones allegation” insufficient to establish disclosure of limitation); *Sisvel Int’l S.A. v. Sierra Wireless, Inc.*, 82 F.4th 1355, 1364 (Fed. Cir. 2023) (rejecting combination that was “too conclusory, lacked clarity, or ... both”). Nor can Petitioner cure this

shortfall later as the rules governing *inter partes* review require that a Petitioner “make [its] case in [the] petition.” *Intelligent Bio-Sys., Inc. v. Illumina Cambridge, Ltd.*, 821 F.3d 1359, 1369 (Fed. Cir. 2016).

Because Petitioner has failed to prove what in Buck, Wilt, or the Buck/Wilt combination satisfies the “user interaction stream” limitation, the Petition fails to demonstrate a reasonable likelihood of prevailing as to any claim.

b. *Petitioner Fails To Demonstrate That Buck/Wilt Discloses Or Renders Obvious A “User Interaction Stream.”*

Even if we were to assume, *arguendo*, that Petitioner’s identification of a “user interaction stream” were sufficiently clear, Petitioner still fails to show that these disparate teachings qualify as a “user interaction stream.” Petitioner, again, essentially points to the fact of user inputs via, *e.g.*, a keyboard, and Buck’s disclosure of a “CPU instruction stream.” Pet., 34. This, however, is not shown to be a “user interaction stream.”

To the extent Petitioner maintains that the mere fact of user commands entered via hardware input devices such as a keyboard, a pointing device such as a mouse, trackball, or touchpad, or a microphone inherently disclose a “user interaction stream,” this is not shown. Nothing in Buck says these inputs form a “user interaction stream.” Instead, the only “stream” Petitioner points to is Buck’s discussion of a “CPU’s instruction stream” in paragraph [0008]. Pet., 34. But

Petitioner does not offer any explanation of why a CPU instruction stream would qualify as a “user interaction stream.” Indeed, ***Petitioner does not even show*** that the user inputs have anything to do with the “CPU instruction stream.”

To the extent Petitioner relies on inherency, it has not shown that the high standard for inherency is met. Inherency requires that the unstated element necessarily be present. *Par Pharm., Inc. v. TWi Pharms., Inc.*, 773 F.3d 1186, 1195–96 (Fed. Cir. 2014) (“A party must, therefore, meet a high standard in order to rely on inherency to establish the existence of a claim limitation in the prior art in an obviousness analysis—***the limitation at issue necessarily must be present***, or the natural result of the combination of elements explicitly disclosed by the prior art.”). Neither Petitioner nor Dr. Rosing attempts to show that user inputs necessarily form or are implemented via a “user interaction stream.” Ex. 2030 [Khatri-Decl.] ¶ 66. Nor has it shown that the “CPU’s instruction stream” necessarily constitutes a “user interaction stream.”

Petitioner, again, cites Buck’s paragraph [0008] (Pet., 34) as allegedly disclosing that the CPU has streams, but that paragraph, as Dr. Khatri explains, says nothing about a “user interaction stream” and merely refers to the “CPU’s instruction stream” in a co-processor context:

Buck’s paragraph [0008] states: “Co-processors are not a new feature of personal computers. The integral floating point co-processor

shares the host CPU's instruction stream and has access to CPU memory.” Ex. 1004 [Buck] [0008]. As can be seen, this is general background information that is not specific to Buck’s inventive system and refers to the “***CPU’s instruction*** stream,” not a “***user interaction*** stream,” as the claims require. Nothing in this paragraph suggests that the “CPU instruction stream” is a “user interaction stream” and Petitioner does not explain why the two streams should be deemed synonymous. Petitioner has not even shown that any user inputs would form a part of or have anything to do with this CPU instruction stream.

Petitioner next states (Pet., 34), “[a]s the ‘host’ processor, Buck’s CPU receives and interprets ‘commands and information’ from user interactions through input interface 160,” again citing Dr. Rosing’s testimony at paragraphs 167–68. Dr. Rosing’s declaration at paragraph 167–168 does not provide any more information indicating that Buck discloses a “user interaction stream.”

Quite simply, Petitioner in no way shows that a “CPU instruction stream” is necessarily a “user interaction stream” even if we were to assume, *arguendo*, that the CPU instruction stream may include some elements influenced by a user’s input.

Ex. 2030 [Khatri-Decl.] ¶¶ 67–69.

Ultimately, Petitioner assumes that Buck’s reference to user inputs and a disconnected discussion of a “CPU instruction stream” suffices to disclose a “user interaction stream”—but never demonstrates that this is so. At best, Petitioner

relies on an implicit, unproven theory of inherency, which is insufficient to support institution.

c. *Petitioner Fails To Demonstrate That Any Alleged “User Interaction Stream” Performs The Claimed Functions.*

Petitioner also fails to show that any alleged “user interaction stream” performs the functions the claims require. That failure compounds across multiple limitations that expressly invoke the “user interaction stream” and assign it specific responsibilities— including controlling data transfers, coordinating with the computational stream, and handling user commands during execution.

For example, limitation 21[b] requires “data exchange between the user interaction stream and the computational stream,” with the computational stream “controlling” that exchange. But in its discussion of 21[b], Petitioner identifies no such exchange between two streams. Instead, it asserts that “Buck’s CPU (or a particular thread of the CPU interacting with the GPU)” “remains in a state of suspension” and concludes just from that fact that the “computational stream” handles the relevant functionality. Pet., 40. The claim, however, does not refer to a CPU or a CPU thread—it requires coordination between the user interaction stream and the computational stream with the former controlling the exchange. The prior art simply does not teach this limitation.

Limitation 21[c] requires “shifting control of a data exchange between the

user interaction stream and the computational stream to the computational stream.” Yet here too, Petitioner cites only to its prior discussion of 21[a] and 21[b], and concludes that in the supposed combination of Buck and Wilt, the “CPU” is prevented from “exchanging data with the GPU.” Pet., 42–43. Again, Petitioner substitutes the CPU for the claimed stream and fails to demonstrate any actual shift in control between the streams as recited.

Limitation 21[d] similarly requires “shifting control of the data exchange between the user interaction stream and the computational stream to the user interaction stream.” Petitioner’s analysis again rests on its discussion of 21[b] and 21[c], conclusorily relying on Buck’s “CPU (or a particular thread of the CPU),” without identifying any user interaction stream or showing that it assumes control of data exchange after computation. Pet., 44.

The same pattern recurs in Petitioner’s treatment of limitations 21[e]–[f] and 33[f], which require that the “user interaction stream” receive user commands or interactions and that those commands/interactions be executed at times determined by the computational stream to avoid data corruption. For 21[e], Petitioner again points back to its discussion of 21[a] and asserts in a conclusory manner, without further analysis, that “Buck discloses that user commands received through its user input interface coupled to the CPU are received by the user interaction stream executed by the CPU.” Pet., 45. But Petitioner’s discussion of 21[a] never

discusses a “user interaction stream” receiving a user command. Rather, Petitioner only says that “Buck’s CPU processes these user interactions and commands by executing a user interaction stream” (*id.*, 34) and argues that “[a] POSITA would have understood that ***Buck’s interactive system*** for simulating artificial neural networks would receive a variety of inputs to the artificial neural network, including commands and information.” *Id.*, 35. This does not remotely show that the “user interactive stream” receives user commands. Petitioner’s discussion of 33[f] merely cites to the discussion of 21[e]–[f]. *Id.*, 78.

Petitioner’s theory becomes even less clear when it appears to shift from Buck to Wilt as satisfying the “user interaction stream.” In its discussion of limitations 21[b] and 21[e], Petitioner suddenly suggests that Wilt—not Buck—discloses a “user interaction stream.” The Petition states: “Like the ‘computational stream’ in the ’461 patent, Wilt’s command buffer controls whether data exchange takes place between the user interaction stream executed by the CPU and the computational stream executed by the GPU in the same way the ’461 patent describes that feature in the claims.” Pet., 41 (citing Ex. 1001 [’461] 8:14–9:52 and Ex. 1003 [Rosing-Decl.] ¶¶ 182–84). Petitioner offers no citation to Wilt for this alleged “user interaction stream.” The cited declaration paragraphs do not identify any such stream either; at most, they refer to a “token stream” without explanation as to why that might be a “user interactive stream.” See Ex. 1003

[Rosing-Decl.] ¶¶ 182–84. The Petition adds that “when Wilt’s CPU receives a command for the GPU, it writes that command into a command buffer.” Pet., 45. But again, that is a generic statement about CPU behavior, not a mapping to a stream that performs the specific functions the claims require.

Ultimately, Petitioner has not shown that any alleged “user interaction stream”—in Buck, Wilt, or the combination—performs the data exchange, control-shifting, or command-handling roles required by the claims. Instead, Petitioner shifts between Buck and Wilt and further, between a CPU, a CPU thread, a command buffer, and vague assertions of functionality without ever identifying a structure that performs as claimed. Accordingly, Petitioner fails to demonstrate that limitations 21[a], 21[b], 21[c], 21[d], 21[e]–[f], 33[d], and 33[f] are disclosed or rendered obvious.

2. Petitioner Fails To Consistently Identify What In Buck/Wilt Satisfies The “Computational Stream” Limitations (All Claims, All Grounds).

The claims also repeatedly invoke a “computational stream”—highlighted in blue below—across nearly every limitation:

21. A method of executing computations representing an artificial neural network on a computer system comprising at least one central processing unit (CPU), a processing unit, a first memory partition, and a second memory partition, the method comprising: executing, by the at least one CPU, a user interaction stream, the user interaction stream controlling transfer of inputs to the artificial neural network to the first memory partition and the second memory partition; executing, by the processing unit, a computational stream, the computational stream controlling data exchange between the user interaction stream and the computational stream during execution of the computations representing the artificial neural network; shifting control of a data exchange between the user interaction stream and the computational stream to the computational stream in response to starting execution of the computations representing the artificial neural network; shifting control of the data exchange between the user interaction stream and the computational stream to the user interaction stream in response to completion or interruption of the computations representing the artificial neural network; queueing a user command received by the user interaction stream during execution of the computations representing the artificial neural network; and executing the user command during execution of the computations representing the artificial neural network at times determined by the computational stream.	33. A system for executing computations representing an artificial neural network, the system comprising: a camera to acquire input data for the artificial neural network; a first memory partition; a second memory partition; at least one central processing unit (CPU), operably coupled to the camera, the first memory partition, and the second memory partition, to execute a user interaction stream, the user interaction stream controlling transfer of the input data acquired by the camera to the first memory partition and the second memory partition during execution of the computations representing the artificial neural network; a processing unit, operably coupled to the first memory partition, the second memory partition, and the at least one CPU, to execute a computational stream, the computational stream controlling transfer of the input data from the first memory partition and the second memory partition during execution of the computations representing the artificial neural network, the execution of the computations representing the artificial neural network occurring while the camera is acquiring the input data; and a controller, operably coupled to the at least one CPU and the processing unit, to queue user interactions received by the user interaction stream during the execution of the computations representing the artificial neural network for performance at times selected to avoid data corruption.
--	---

Yet, mirroring the same problems present in its analysis of the “user interaction stream,” Petitioner fails to consistently identify what in the Buck/Wilt combination constitutes the “computational stream.” Its position suffers from three distinct flaws. First, as with the “user interaction stream,” Petitioner never clearly and consistently identifies what it relies on in the prior art as satisfying the “computational stream.” See Section II.B.2(a). Second, the various candidates it does invoke—Buck’s computations, Wilt’s command buffer(s)³, and Wilt’s

³ Petitioner refers interchangeably to Wilt’s “command buffers” and its

scheduler—do not, individually or collectively, perform the full range of required functions. *See* Section II.B.2(b). Third, Petitioner improperly and without explanation maps the same structure to multiple distinct claim terms, including both the “computational stream” and the “controller,” in violation of basic rules of claim differentiation. *See* Section II.B.2(c). Each of these deficiencies is sufficient reason to deny institution.

a. *Petitioner Fails To Clearly And Consistently Identify A “Computational Stream.”*

Petitioner fails to clearly and consistently identify what it is relying on as the “computational stream.” Despite the term appearing in nearly every limitation, Petitioner shifts its identification at least three times, without ever reconciling the discrepancies.

The “computational stream” first appears in limitation 21[b], which requires a “executing, by the processing unit, a computational stream, the computational stream controlling data exchange between the user interaction stream and the computational stream during execution of the computations representing the artificial neural network.” Petitioner first points to Buck’s GPU as the “processing unit” and asserts that it executes a computational stream. Pet., 39. But Petitioner

“hardware-specific commands.” *E.g.*, Pet., 16.

does not specify what in Buck constitutes the stream itself. *Id.* Instead, Petitioner states:

Buck discloses a processing unit executing a computational stream including computations representing an artificial neural network. Ex1004, [0008]; Ex1003, ¶177. Buck’s GPU (a “processing unit”) is an autonomous processor with its “own instruction stream[], datapaths, and dedicated memory”... These operations include computations to train and test a two-layer artificial neural network.

Pet., 39. Thus, we are told that Buck executes a computational stream but we are not told what it is.

Petitioner points to Buck’s paragraph [0008] as satisfying the “computational stream” (Pet., 39), but this is the same paragraph Petitioner relied on as satisfying the CPU’s “user interaction stream” (*id.*, 34). While Buck references both a “CPU’s instruction stream” and a “GPU[] instruction stream[],” Petitioner never explains why one would qualify as a “user interaction stream” and the other as a “computational stream.” The claims treat the “user interaction stream” and the “computational stream” as distinct and with defined and differing roles; Petitioner offers no reasoning for treating generic instruction streams as satisfying either or both of those roles. Dr. Rosing, in paragraph 177, again cites to Buck at [0008], using the same argument about the “computational stream” as was made in the Petition. Ex. 1003 [Rosing-Decl.] ¶ 177.

Elsewhere, Petitioner appears to shift its theory again, stating that “the computations representing the simulation of the artificial neural network (the computational stream)” satisfy the limitation. Pet., 40. But this conflates the “computational stream” with the “computations” it is required to execute. Because “computational stream” and “computations” are both claim terms, Petitioner’s statement contravenes the presumption that different claim terms carry different meanings. *See Bd. Of Regents v. BENQ Am. Corp.*, 533 F.3d 1362, 1371 (Fed. Cir. 2008) (“Different claim terms are presumed to have different meanings.”) (citing *CAE Screepnplates Inc. v. Heinrich Fiedler GmbH & Co.*, 224 F.3d 1308, 1317 (Fed. Cir. 2000)).

Petitioner then shifts yet again—this time from Buck to Wilt—asserting: “Wilt also discloses a GPU that executes a stream of hardware-specific commands that together constitute a ‘computational stream.’” Pet., 40. But Petitioner never clarifies how this “stream of hardware-specific commands” relates to or supplants the earlier references to Buck’s instruction streams or computations.

Petitioner does not reconcile these shifting theories and never identifies which of Buck’s GPU instruction stream, Buck’s neural network computations, or Wilt’s hardware-specific commands constitutes the computational stream in the combined system. This lack of clarity fails to meet Petitioner’s burden to “identify with particularity” how the prior art satisfies the claim limitation. *See* 35 U.S.C. §

312(a)(3); *Netflix, Inc. v. DivX, LLC*, 84 F.4th 1371, 1377 (Fed. Cir. 2023)

(“Ultimately, it is the petitioner’s burden to present a clear argument.”).

b. *None Of Petitioner’s Options Perform The Required Functions.*

Even setting aside the identification problem, Petitioner also fails to show that any of the inconsistent and varying options it identifies perform the claimed functions of the “computational stream.”

Limitation 21[f], for example, requires “executing the user command during execution of the computations representing the artificial neural network at times determined by the computational stream.” Petitioner never contends that any of Buck’s instruction stream or computations or Wilt’s hardware-specific commands make that timing determination. Instead, it points to an entirely different structure—Wilt’s scheduler that makes the priority “decision” as to when a “command buffer” is “submit[ted]” “for execution”:

Wilt’s scheduler then submits the command buffer for execution at a particular point within the computational stream. *Id.*, [0107]... *the scheduler’s decision* is based on the status and priority of the other command buffers.

Pet., 46–47; *see also* Ex. 1003 [Rosing-Decl.] ¶ 204. Thus, Petitioner assigns the claimed time determination role not to the “computational stream,” but to a separate component, the scheduler.

Petitioner also fails to show that any identified structure satisfies limitation 33[e], which requires that “the *computational stream control* transfer of the input data... during execution of the computations.” In addressing this limitation, Petitioner first states that the “command buffer controls the data exchange between the user interaction stream and the computational stream *by requesting the inputs*.” Pet., 77 (citing Ex. 1003 [Rosing-Decl.] ¶¶ 267–270, 291). Petitioner, however, cites nothing in Wilt in support of this contention. Petitioner further does not explain why “requesting” an input is the same thing as “controlling transfer of the input data from the first memory partition and the second memory partition during execution of the computations representing the artificial neural network.” The portions of Dr. Rosing’s declaration that are cited (paragraphs 267–270 and 291) similarly confuse “requesting” an input with “controlling” the transfer of input data.

Even if that could be overlooked, Petitioner’s argument gets more confusing from there as Petitioner then argues that it is “[t]he scheduler”—not the command buffer—that “can then facilitate the transfer of this input data”:

Wilt’s command buffer controls the data exchange between the user interaction stream and the computational stream by requesting the inputs. Ex1003, ¶¶267–70, 291. Specifically, the completion or interruption of a command from the command buffer causes the GPU to provide a notification to the CPU, which can then cause a user to be

prompted for input data. *Id.* ***Wilt’s scheduler*** makes resources and data available according to the requirements of the command buffers (part of the computational stream). Ex1005, [0024]. ***The scheduler can then facilitate the transfer of this input data*** from the CPU (first memory partition) to a segment of the GPU memory (second memory partition). Ex.1003, ¶291.

Pet., 77. Petitioner fails to explain how the “command buffer” “controls” the transfer when it is “the scheduler” that “facilitate[s] the transfer of th[e] input data.” Even if that could be overlooked, Petitioner does not show that either Wilt’s command buffer or Wilt’s scheduler performs this control “during execution of the computations,” as the claim requires.

These gaps are significant. A “computational stream” that neither controls the timing of user interaction execution nor governs the required data transfers is not the claimed “computational stream.” Petitioner’s failure to show that any identified structure performs these core functions is independently fatal to its obviousness case.

c. *Petitioner Improperly Equates The “Computational Stream” With The “Controller.”*

Petitioner’s internal inconsistencies are most apparent in its treatment of limitation 33[f], which requires that a separate “controller” determine when to perform user interactions. Here, Petitioner asserts: “The Buck/Wilt scheduler, which is operably coupled to the CPU and GPU, is a controller.” Pet., 78.

But Petitioner’s reliance on Wilt’s scheduler as the “controller” directly contradicts Petitioner’s earlier reliance on that same scheduler as part of the “computational stream.” *See* Pet., 77; Ex. 1003 [Rosing-Decl.] ¶ 291. Petitioner earlier used Wilt’s scheduler for the “computational stream[s]” recited in 21[f] and for 33[e]. (Pet., 46–47, 77). Petitioner thus assigns the same structure—Wilt’s scheduler—to two distinct claim terms: the “computational stream” and the “controller.” But Petitioner’s approach violates the fundamental presumption that “[d]ifferent claim terms are presumed to have different meanings.” *See BENQ Am.*, 533 F.3d at 1371; *see also Aspex Eyewear, Inc. v. Marchon Eyewear, Inc.*, 672 F.3d 1335, 1349 (Fed. Cir. 2012) (“The fact that the two adjacent claims use different terms in parallel settings supports the ... conclusion that the two terms were not meant to have the same meaning.”).

Petitioner offers no explanation for why Wilt’s scheduler qualifies as both a “controller” and the “computational stream.”

3. Petitioner Fails To Show “The Computational Stream Controlling Data Exchange” Between The Streams (Claims 21–32, 38–41 All Grounds).

Limitation 21[b] requires “the computational stream *controlling data exchange* between the user interaction stream and computational stream during execution of the computations representing the artificial neural network.” In its attempt to satisfy this limitation, however, Petitioner relies upon scenarios where

no data is exchanged. First, Petitioner relies upon a scenario where Buck’s CPU is “blocked” and, consequently, there is no “data exchange” between any alleged “user interaction stream” (of the CPU) and any “computational stream.” *See* Section II.B.3(a). Second, Petitioner points to scenarios wherein Wilt can only execute one command buffer at a time. This too, however, in no way controls “data exchange” between the streams. *See* Section II.B.3(b). Third, Petitioner relies upon Wilt’s “preemption” wherein its GPU may preempt itself and cease work. But, here too, the GPU ceasing work in no way shows a “computational stream” controlling data exchanges between the streams. *See* Section II.B.3(c). By solely pointing to scenarios where no data is exchanged between any alleged streams, Petitioner never demonstrates that any alleged “computational stream” controls data exchanges between the streams as required by limitation 21[b].

a. *Petitioner Fails To Prove That Buck Discloses The Computational Stream Controlling Data Exchange During Execution Of Computations.*

Limitation 21[b], again, requires “the computational stream controlling data exchange between the user interaction stream and the computational stream during execution of the computations representing the artificial neural network.”

Petitioner argues that “Buck/Wilt discloses this limitation,” first focusing on Buck alone. Pet., 38. Petitioner points to a scenario wherein “Buck’s GPU begins executing a simulation of the artificial neural network[.]” *Id.*, 40. During that

simulation, Petitioner makes clear that it relies upon a scenario wherein Buck's CPU is blocked and "prevented from providing instructions or data to the GPU":

Buck's CPU (or a particular thread of the CPU interacting with the GPU) "blocks," and remains in a state of suspension until the GPU completes executing the simulation. Ex1004, [0235], [0211]–[0250]; Ex1003, ¶180. ***Buck's blocked CPU, thus, is prevented from providing instructions or data to the GPU, perform synchronization with the GPU, or otherwise exchange data with the GPU*** to affect the computations representing the simulation of the artificial neural network (the computational stream). Ex1003, ¶¶180–81. In other words, when Buck's GPU begins computations, the computational stream controls data exchange data exchange [*sic*] between the user interaction stream executed by the CPU and the computational stream executed by the GPU while the computations are executed ***by causing suspension of Buck's CPU***. *Id.*

Pet., 39–40. As seen above, Petitioner's argument is essentially that because Buck's CPU is blocked and in a "state of suspension until the GPU completes executing the simulation," "Buck's computational stream controls data exchange in the way described in the '461 patent." *Id.*

Petitioner's theory does not withstand scrutiny. A computational stream cannot "control" a data exchange between itself and the user interaction stream if no such exchange exists or is even possible. As Dr. Khatri explains:

When the CPU blocks, per Petitioner and Dr. Rosing,^[4] there is no data exchange between the CPU and the GPU. Pet., 39–40. I do not understand any sense in which the GPU—much less a computational stream—can be said to “control data exchanges” with the CPU’s “user interaction stream” when the CPU is blocked and cannot be controlled, and there is thus no data to be exchanged, or that can be exchanged.

Indeed, Petitioner’s explanation of “block[ing]” confirms that there simply is no data exchange, as Petitioner argues that “Buck’s blocked CPU ... *is prevented from providing* instructions or *data to the GPU ... or otherwise exchange data with the GPU.*” *Id.*, 40. Indeed, in Petitioner’s worldview, the GPU and CPU are *not* communicating at this time, since per Petitioner, in the blocked state, the CPU cannot “*perform synchronization with the GPU.*” *Id.* Notably missing from Petitioner’s discussion is any mention of data being exchanged outside of the CPU’s control or how Buck’s alleged computational stream controls the data exchange, as the claims require. Petitioner’s logic suggests that it is actually the CPU that controls data exchange between the streams during computations, which is exactly the converse of what limitation 21[b] requires. Petitioner’s logic simply does not follow.

⁴ While Buck does not describe what “block[ing]” entails, Buck appears to imply that it only means that the CPU cannot “compute” while blocked. Ex. 1004 [Buck] [0235] (“In this example, the CPU gives instruction to the GPU and blocks until the GPU complete these instructions”).

Ex. 2030 [Khatri-Decl.] ¶¶ 95–96.

Because Petitioner identifies no data exchange occurring during execution—and no mechanism by which the computational stream controls such an exchange—Petitioner fails to show that Buck discloses the required limitation.

b. *Petitioner Fails To Show That Wilt Discloses The Limitations.*

Next, Petitioner focuses upon Wilt alone, but again fails to explain how it relates to Buck or satisfies limitation 21[b]. Petitioner’s theory is that because Wilt’s GPU executes one command buffer at a time, this somehow means that it “controls whether data exchange takes place between the user interaction stream executed by the CPU and the computational stream executed by the GPU”:

Wilt explains that a “graphics processor can only execute one command buffer at a time.” Ex1005, [0086], [0108]. Once the GPU begins executing a command buffer, “the command buffer remains in the running state...until some event preempts execution of the command buffer or *the entire contents of the command buffer have been read and executed.*” *Id.* While Wilt’s GPU is executing a command buffer (a portion of the computational stream), the GPU does not execute any other command buffer and thus cannot change the instructions it executes or the data on which it operates while it is executing a command buffer. Ex1003, ¶¶182–84. Like the “computational stream” in the ’461 patent, Wilt’s command buffer controls whether data exchange takes place between the user interaction

stream executed by the CPU and the computational stream executed by the GPU in the same way the '461 patent describes that feature in the claims. *Id.*; Ex1001, 8:14–9:52.

Pet., 41.

But this suffers from the same core flaw as Petitioner's argument under Buck. Petitioner relies on a scenario in which the system is effectively locked down until “*the entire contents of the command buffer have been read and executed.*” Wilt's GPU executes a single command buffer at a time and does not process other commands or data. This ability to execute one command buffer at a time is a property of Wilt's system; it does *not* teach shifting control of a data exchange between a “user interaction stream” and a “computational stream.” There is no discussion or identification of any mechanism by which any data exchange occurs between the user interaction stream and the computational stream during execution of the command buffer—much less one that is controlled by the computational stream.

Indeed, Petitioner does not point to any data exchange at all in this scenario. It merely describes a command buffer being executed in isolation, without interruption, input, or coordination with any other stream. And if no data exchange is occurring, then there is nothing for the computational stream to control. The limitation is not met.

c. *Petitioner's Preemption Theory Also Fails.*

Petitioner's next theory focuses upon Wilt's discussion of preemption. Specifically, Petitioner argues that in preemption, Wilt's GPU may halt execution of one of its command buffers to execute the preempting command buffer and that this, somehow, discloses the computational stream controlling data exchanges with the CPU's user interaction stream:

Wilt's discussion of preemption additionally teaches that its computational stream controls whether data exchange takes places [*sic*]. Ex1005, [0108], [0127]. Preemption is the reclaiming of a resource from a process before it has finished using the resource. Ex1003, ¶¶185–86. When a currently executing task is preempted, it is interrupted so that another task, typically with a higher priority, can be executed. *Id.* The priorities of various command buffers, which are portions of the computational stream, are attributes of the computational stream. *Id.* Based on these attributes of the computational stream, ***Wilt's GPU may halt execution of the preempted command buffer to execute the preempting command buffer and change the instructions it executes or the data on which it operates.*** *Id.* In this additional way, Wilt teaches that the computational stream controls whether data exchange takes place between the user interaction stream and the computational stream at least in the same way the '461 patent describes that feature in the claims. *Id.*

Pet., 41–42.

This is just a variation on the same failed logic. As with Buck’s blocked CPU and Wilt’s single-command buffer execution, Petitioner once again points to a scenario where no data exchange takes place whatsoever between any user interaction stream and a computational stream. Wilt’s GPU preempting one command buffer to run another does not involve any interaction with the CPU, and certainly does not constitute data exchange between the user interaction stream and computational stream—let alone control over that exchange. In fact, all that occurs is that a higher priority command buffer (which has no control dependency over the one it preempts) is run in place of a lower priority command buffer. Ex. 2030 [Khatri-Decl.] ¶ 103.

At most, Petitioner shows that Wilt’s GPU determines which command buffer to execute. That has nothing to do with whether the computational stream controls transfer of information from the CPU or user interaction stream. There is no evidence that any data exchange is even possible in this context, and no suggestion that any such exchange is governed by the computational stream. As Dr. Khatri explains:

There is no reason to believe that Wilt’s preemption teaches that any of the alleged computational streams Petitioner identifies controls data exchange between the user interaction stream and computational stream. Wilt merely describes its command buffer as “preemptible,” but does not describe what entity actually makes the preemption

decision. *See* Ex. 1005 [Wilt] [0110], [0127]. At most, Wilt describes that “the hardware may have to preempt [the command buffer’s] execution,” but this at most refers to the GPU (which is hardware), not the previously alleged computational streams (which are software) (identified by Petitioner as Wilt’s command buffer or the scheduler). *Id.*, [0127]. In accord, Petitioner concedes that “**Wilt’s GPU** may halt execution of the preempted command buffer,” not any of the alleged computational streams. Pet., 42.

Ex. 2030 [Khatri-Decl.] ¶ 102.

Petitioner also provides no evidence to support its contention that a command buffer is ever preempted in order to “change the instructions it executes or the data on which it operates.” *See* Pet., 42. Petitioner simply cites Dr. Rosing’s conclusory and unsupported testimony. Ex. 1003 [Rosing-Decl.] ¶ 185. As an initial matter, it is unclear how, even if this were true, it would in any way show a computational stream controlling data exchange with a user interaction stream.

Additionally, it is not shown to be true. Wilt does not describe preemption as a mechanism for modifying command content. Instead, it identifies external events, such as VSYNC or hardware enforcement of a time quantum, as the causes for preemption: “a variety of events could potentially cause preemption of a command buffer during execution, ranging from an external event such as VSYNC

to hardware enforcement of a time quantum.” Ex. 1005 [Wilt] [0127]. Preemption is simply a mechanism to choose to run a higher priority command buffer instead of a lower priority one. Ex. 2030 [Khatri-Decl.] ¶¶ 103–105.

These events enable the system to reallocate and re-prioritize GPU time—not to alter what is in a specific command buffer or introduce mid-execution communication with the CPU. In fact, Wilt explains that the preempted command buffer is saved so that it can resume execution from the point of preemption, without needing any CPU intervention: “if the command buffer is preempted, it is assumed that the hardware has the capability to save the hardware context in such a way that execution of the command buffer can be resumed where the preemption occurred.” Ex. 1005 [Wilt] [0110].

That is the opposite of dynamic data exchange or controlled transfer. Petitioner has not shown that any “computational stream” controls anything relevant to the claimed limitation, only that a GPU scheduler enforces task switching between GPU tasks based on unrelated events “such as VSYNC [or] hardware enforcement of a time quantum.” Ex. 1005 [Wilt] [0127].

Accordingly, this theory, like the others, fails to show that the computational stream controls data exchange between the streams during execution.

4. Petitioner Fails To Demonstrate That The “User Interaction Stream Controlling Transfer Of The Input Data” To The Memory Partitions “During Execution Of The Computations” (Claims 33–37, All Grounds).

Limitation 33[d] requires “the user interaction stream controlling transfer of the input data ... to the first memory partition and the second memory partition *during execution of the computations representing the artificial neural network.*” Petitioner fails to prove that this limitation is met. In fact, its position is directly at odds with its own argument under limitation 21[b].

To satisfy limitation 33[d], Petitioner argues that during the GPU’s computations, the CPU “can continue to receive and store input data” and “transfer[] the data to the GPU”:

Buck/Wilt’s resource management system “facilitate[s] the host microprocessor’s performance of useful work while the coprocessor hardware is running.” Ex1005, [0114]. Therefore, Buck/Wilt’s *CPU can continue to receive and store input data generated by the video camera while the GPU is executing computations.* Ex1003, ¶284. Because this data is used as input to the Buck/Wilt artificial neural network, *the CPU transfers the data to the GPU* while the GPU continues to perform computations representing the artificial neural network. *Id.*; Ex1007, 5.

Pet., 75–76.

But this theory contradicts Petitioner’s argument under claim 21[b], which requires “the computational stream controlling data exchange between the user interaction stream and the computational stream during execution...” and where Petitioner argues that Buck satisfies this limitation because ***the CPU is blocked while the GPU runs***: “[T]he CPU... is prevented from providing instructions or data to the GPU, perform synchronization with the GPU, or otherwise exchange data with the GPU.” *See* Pet., 40; *see also* Section II.A.3(a). ***Petitioner’s theory for 21[b] thus rests on the premise that the user interaction stream is not transferring data to the GPU during execution.*** Yet in its argument for 33[d], Petitioner contends that the same stream is doing exactly that.

Moreover, Petitioner explicitly incorporates its analysis of 21[b] in its discussion of claim 33, in limitation 33[e]. Pet.,76 (“*See* 21[pre], 21[b], and 21[c]...”). Yet Petitioner makes no attempt to reconcile the claim that the CPU is “prevented from providing... data to the GPU” with its simultaneous claim, in the same section of the Petition, that the CPU is actively transferring input data to the GPU during execution. Petitioner cannot have it both ways. It cannot assert that the user interaction stream is transferring data during execution while simultaneously relying on a theory that requires the CPU to be blocked and thus ***unable*** to transfer data. It offers no alternative embodiment, no fallback position, and no effort to resolve the inconsistency.

For this additional reason, Petitioner’s argument as to claims 33–37 should be rejected.

5. Petitioner Fails To Demonstrate That The “User Command” Limitation Is Disclosed And/Or Obvious (All Claims, All Grounds).

Limitation 21[e]–[f] requires “a user command received by the user interaction stream during execution of the computations representing the artificial neural network” and “executing the user command *during execution* of the computations ... at times determined by the computational stream.” Similarly, limitation 33[f] requires “a controller ... to queue user interactions received by the user interaction stream *during the execution of the computations* representing the artificial neural network for performance at times selected to avoid data corruption.”

Thus, the claims require a user command/interaction that the CPU’s user interaction stream receives to be executed during the computations representing the artificial neural network, which are executed on the GPU. Petitioner argues that “Buck/Wilt discloses” these limitations. Pet., 45–47, 78–80. But Petitioner never presents a coherent theory that ties the required functionality to any structure.

Petitioner’s argument begins by invoking its analysis of claim 21[a], asserting that Buck’s user input interface (keyboard, mouse) allows user commands to be received by the CPU. See Pet., 45 (citing Ex. 1004 [Buck]

[0049]–[0050]; Ex. 1003, ¶¶165–75, 197). Petitioner then shifts to Wilt, asserting that:

When Wilt’s CPU receives a command for the GPU, it writes that command into a command buffer. Ex1005, [0097]–[0106]. Once the command buffer is full or the command has been completely written into the command buffer, the command buffer enters a waiting state until various waiting criteria are fulfilled. *Id.*

Pet., 45, 78 (incorporating limitation 21[e] which incorporates the limitation 21[a] discussion).

But this does not come close to satisfying the actual limitations. First, Petitioner never shows that Buck’s commands are received “during execution” of the neural network computations. Buck merely states that “a user may enter commands and information into the computer 110 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad.” Ex. 1004, [0049]. Buck does not identify what these commands are for, when they are received, or whether they are processed at any particular point in the GPU’s execution.

Second, Petitioner never identifies a user command intended for the GPU that is received *during execution* of the computations representing the artificial neural network and then executed at a time *determined by the computational stream*. At most, Petitioner argues that commands are received by the CPU (via

Buck) and then executed by the GPU based on scheduling rules or buffer state (via Wilt). But that is not what the claims require. The user command must be received by the user interaction stream during execution, and its execution timing must be controlled by the computational stream (limitation 21[f]) or queued “for performance at times selected to avoid data corruption” (limitation 33[f]).

Petitioner fails to bridge that gap. It does not argue that any user command is (1) directed to the GPU, (2) received during execution of the computations, and (3) executed in coordination with or under the control of the computational stream. It simply assumes that these steps occur, without mapping them to any structure or combination of Buck and Wilt.

As a result, Petitioner has failed to show that this limitation is disclosed or rendered obvious under any asserted ground.

III. CONCLUSION

For the foregoing reasons, Patent Owner respectfully requests that institution be denied under 35 U.S.C. § 314(a).

Respectfully submitted,

/ Kenneth J. Weatherwax /

Kenneth J. Weatherwax (Reg. No. 54,528)

Nathan Lowenstein, *pro hac vice*

Colette Woo, *pro hac vice*

LOWENSTEIN & WEATHERWAX LLP

Date: July 15, 2025

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITS

This Patent Owner Preliminary Response (the “POPR”) consists of 10,980 words, excluding table of contents, table of authorities, certificate of service, this certificate, or table of exhibits. The POPR complies with the type-volume limitation of 14,000 words as mandated in 37 C.F.R. § 42.24. In preparing this certificate, counsel has relied on the word count of the word-processing system used to prepare the paper (Microsoft Word).

Respectfully submitted,

/ Colette Woo /

Date: July 15, 2025

CERTIFICATE OF SERVICE

The undersigned hereby certifies that the following documents were served
by electronic service, by agreement between the parties, on the date below:

PATENT OWNER PRELIMINARY RESPONSE

EXHIBIT 2030

The names and addresses of the parties being served are as follows:

Brian M. Buroker	bburoker@gibsondunn.com
L. Kieran Kieckhefer	kkieckhefer@gibsondunn.com
Nathan Curtis	ncurtis@gibsondunn.com
Vivian Lu	vlu@gibsondunn.com

Respectfully submitted,

/ Colette Woo /

Date: July 15, 2025