

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Trend Micro, Inc.,

Petitioner

v.

Open Text Inc.,

Patent Owner

Case IPR2023-00662
Patent No. 10,599,844

PATENT OWNER'S PRELIMINARY RESPONSE

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

I.	Introduction.....	1
II.	The Board Should Exercise Its Discretion To Deny Institution Under 35 U.S.C. § 314(A).....	3
	A. Factor 1: The District Court Has Not Granted And Will Not Grant A Stay.	5
	B. Factor 2: A District Court Trial Is Expected Near The Projected Statutory Deadline.	10
	C. Factor 3: The Parties Will Have Invested Significant Resources In The District Court Proceeding.....	12
	D. Factor 4: The Issues And Parties Overlap Significantly.	14
	E. Factor 5: Both Parties Are Parties In The Parallel Litigation	15
	F. Factor 6: The Petition Does Not Present “Compelling Evidence Of Unpatentability.”	17
III.	The Petition Is Not Reasonably Likely To Succeed.....	18
	A. Background.....	20
	1. The State Of The Art	20
	2. The ’844 Patent.....	22
	3. Claim Construction.....	27
	4. The Level Of Ordinary Skill	29
	B. Overview Of The Cited Art.....	29
	1. Overview Of Tahan.....	29
	2. Overview Of Menahem.....	38
	C. Ground 1: The Combination Of Tahan And Menahem Did Not Render Claims 1-5, 8, 10-14, And 17 Obvious.....	39
	1. The Cited Art Fails To Suggest Turning Features On Or Off At The Classification Stage.	40
	2. The Cited Combination Does Not Extract Static Data	

	Points That “Represent Predefined Character Strings.”	53
D.	Ground 2: The Combination Of Tahan, Menahem, And Schultz Did Not Render Dependent Claims 6, 7, 9, 15, 16, 18, And 20 Obvious.	60
IV.	There Are Objective Indicia Of Non-Obviousness.....	60
V.	Conclusion	62

TABLE OF AUTHORITIES

Cases:

<i>Abiomed, Inc., et al., v. Maquet Cardiovascular, LLC,</i> IPR2017-01204, Paper 9 (PTAB Oct. 23, 2017)	50
<i>Advanced Bionics, Oticon Medical AB v. Cochlear Ltd.,</i> IPR2019-001469, Paper No. 6 (February 13, 2020)	55
<i>Apple Inc. v. Fintiv, Inc.,</i> IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020)	<i>passim</i>
<i>Apple Inc. v. Fintiv, Inc.,</i> IPR2020-00019, Paper 15 (PTAB May 13, 2020)	10-11
<i>Commscope Techs. v. Dali Wireless, Inc.,</i> IPR 2022-01242, Paper 23 (PTAB Feb. 27, 2023)	17
<i>Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.,</i> 800 F.3d 1375 (Fed. Cir. 2015)	39
<i>Eli Lilly & Co. v. Teva Pharms. Int'l GmbH,</i> 8 F.4th 1331 (Fed. Cir. 2021)	49, 58
<i>Ericsson Inc. v. Collision Comms., Inc.,</i> IPR2022-01233, Paper 12 (PTAB Jan. 19, 2023)	14
<i>Graham v. John Deere Co.,</i> 383 U.S. 1 (1966)	61
<i>Harmonic Inc. v. Avid Tech., Inc.,</i> 815 F.3d 1356 (Fed. Cir. 2016)	39
<i>In re Huang,</i> 100 F.3d 135 (Fed. Cir. 1996)	61

Patent Owner’s Preliminary Response
to Petition for IPR2023-00662
Patent No. 10,599,844

<i>Intellectual Ventures I LLC v. Hewlett Packard Enterprise Co.</i> , No. 6:21-cv-00596-ADA, slip op. (W.D. Tex. Dec. 22, 2022)	6
<i>Koninklijke Philips v. Google</i> , 948 F.3d 1330 (Fed. Cir. 2020).....	47, 58
<i>Leo Pharm. Prods., Ltd. v. Rea</i> , 726 F.3d 1346 (Fed. Cir. 2013).....	61
<i>Lumenis Be Ltd. v. BTL Healthcare Techs. A.S.</i> , IPR2021-01280, Paper 8 (PTAB Mar. 9, 2022)	40
<i>Luxshare Precision Industry v. Amphenol Corp.</i> , IPR2022-00132, Paper 10 (PTAB May 3, 2022).....	10
<i>mCom IP, LLC v. Cisco Sys., Inc.</i> , No. 6:22-00261-ADA, slip op. (W.D. Tex. Oct. 20, 2022).....	6
<i>Motorola Mobility LLC v. Ironworks Patents, LLC</i> , IPR2021-00421, Paper 11 (PTAB July 22, 2021)	10
<i>Nintendo Co. v. Gamevice, Inc.</i> , IPR2020-01197, Paper 13 (PTAB Jan. 12, 2021)	10
<i>Nokia of Am. Corp. v. Godo Kaisha IP Bridge</i> , IPR2022-00755, Paper 9 (PTAB Nov. 2, 2022).....	16
<i>NXP USA, Inc. v. Impinj, Inc.</i> , PGR2022-00005, Paper 20 (PTAB Aug. 25, 2022)	11
<i>Regeneron Pharms. Inc. v. Novartis Pharma AG</i> , IPR2020-01317, Paper 15 (PTAB Jan. 15, 2021)	13, 14
<i>Robert Bosch Tool Corp. v. SD3, LLC</i> , IPR2016-01751, Paper 15 (PTAB Mar. 22, 2017)	61

Patent Owner’s Preliminary Response
to Petition for IPR2023-00662
Patent No. 10,599,844

<i>Samsung Elecs. Co. v. Clear Imaging Research LLC</i> , IPR2020-01399, Paper 13 (PTAB Feb. 3, 2021).....	10
<i>Sonrai Memory Ltd. v. Kingston Tech. Co., Inc.</i> , No. 6:21-cv-1284-ADA, slip op. (W.D. Tex. Oct. 18, 2022).....	6
<i>Twinstrand Biosciences, Inc. v. Guardant Health, Inc.</i> , IPR2022-01115, Paper 14 (PTAB Dec. 30, 2022)	60
<i>Vizio, Inc. v. Maxell, Ltd.</i> , IPR2022-01459, Paper 7 (PTAB Apr. 11, 2023).....	16

Statutes:

35 U.S.C. § 312(a)(3).....	40
35 U.S.C. § 314(a)	3, 17

Regulations:

37 C.F.R. § 42.22(a)(2)	40
37 C.F.R. § 42.100(b)	28

Other Authority:

H.R. Rep. No. 112–98, pt. 1, at 40 (2011), 2011 U.S.C.C.A.N.	1
Director’s Memo, Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation, June 21, 2022	3, 4, 11, 12, 17

EXHIBIT LIST

Exhibit	Description
2001	Declaration Of Sam Malek, Ph.D.
2002	Curriculum Vitae of Sam Malek, Ph.D.
2003	<i>Webroot, Inc., Open Text Inc., v. AO Kaspersky Lab et al.</i> , Case No. 6:22-cv-00243-ADA-DTG, Order Granting Third Amended Scheduling Order, Dkt. 142 (W.D. Tex., Dec. 27, 2022)
2004	<i>Sonrai Memory Ltd. v. Kingston Tech. Co. and Kingston Tech. Corp.</i> , Case No. 6:21-cv-1284-ADA, Order Denying Kingston Tech. Co. and Kingston Tech. Corp.’s Opposed Motion to Stay Pending Resolution of <i>Inter Partes</i> Review, Dkt. 94 (W.D. Tex., Oct. 18, 2022)
2005	<i>mCom IP, LLC v. Cisco Systems, Inc.</i> , Case No. 6:22-cv-00261-ADA, Order Denying Defendant’s Motion to Stay Pending <i>Inter Partes</i> Review, Dkt. 42 (W.D. Tex., Oct. 20, 2022)
2006	<i>Intellectual Ventures I LLC et al. v. Hewlett Packard Enterprise Co.</i> , No. 6:21-cv-00596-ADA, Order Denying Hewlett Packard Enterprise Company’s Opposed Motion to Stay Pending Resolution of <i>Inter Partes</i> Review, Dkt. 104 (W.D. Tex., Dec. 22, 2022)
2007	<i>Webroot, Inc. and Open Text Inc., v. AO Kaspersky Lab. et al.</i> , Case No. 6:22-cv-243-ADA-DTG, Claim Construction Order, Dkt. 236 (W.D. Tex. Mar. 16, 2023)
2008	Dani Kass, “Catching Up On Patent Litigation With Judge Albright” (Law360 Mar. 14, 2023); https://www.law360.com/pulse/articles/1582438/print?section=pulse/courts
2009	<i>Webroot, Inc., Open Text Inc., v. AO Kaspersky Lab et al.</i> , Case No. 6:22-cv-00243-ADA-DTG, Order Granting Third Amended Scheduling Order, Dkt. 142 (W.D. Tex., Dec. 27, 2022)
2010	United States Courts, Statistics & Reports, Federal Court Management Statistics—Comparison Within Circuit—During the 12-Month Period Ending June 30, 2022, “Comparison of Districts Within the First Circuit — 12-Month Period Ending June 30, 2022,” available

Patent Owner's Preliminary Response
to Petition for IPR2023-00662
Patent No. 10,599,844

Exhibit	Description
	https://www.uscourts.gov/sites/default/files/fcms_na_distcomparison0630.2022_0.pdf
2011	<i>Webroot, Inc. and Open Text Inc., v. Sophos Ltd.</i> , Case No. 6:22-cv-00240-ADA, Complaint for Patent Infringement, Dkt. 1 (W.D. Tex., Mar. 4, 2022)
2012	<i>Webroot, Inc. and Open Text Inc., v. ForcePoint LLC</i> , Case No. 6:22-cv-00342-ADA-DTG, Defendant ForcePoint LLC's Preliminary Invalidity Contentions (W.D. Tex., Sept. 13, 2022)
2013	<i>Webroot, Inc. and Open Text Inc., v. Sophos Ltd.</i> , Case No. 6:22-cv-00240-ADA-DTG, Defendant Sophos Ltd.'s Preliminary Invalidity Contentions (W.D. Tex., Sept. 13, 2022)
2014	<i>Webroot, Inc. and Open Text Inc., v. Trend Micro Inc.</i> , Case No. 6:22-cv-00239-ADA-DTG, Defendant's Preliminary Invalidity Contentions (W.D. Tex., Sept. 13, 2022)
2015	<i>Webroot, Inc. and Open Text Inc., v. Ao Kaspersky Lab</i> , Case No. 6:22-cv-00243-ADA-DTG, Defendant's Preliminary Invalidity Contentions Appendix A (W.D. Tex., Sept. 14, 2022)
2016	<i>Webroot, Inc. and Open Text Inc., v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.</i> , Case No. 6:22-cv-00241-ADA-DTG, Defendant's Preliminary Invalidity Contentions (W.D. Tex., Sept. 13, 2022)
2017	December 9, 2022 excerpt from discovery hearing transcript

Open Text Inc. ("Patent Owner" or "Open Text") submits this Preliminary Response to the Petition of Trend Micro, Inc. ("Petitioner") seeking *inter partes* review ("IPR") of U.S. Patent No. 10,599,844 ("the '844 patent").

I. INTRODUCTION

The Board should decline to institute trial because an IPR would duplicate the parties' district court litigation efforts and the petition presents weak unpatentability arguments using prior art that fails to disclose key claim limitations.

First, the Board should deny institution under *Fintiv* because institution would undermine the congressional purpose of IPRs to "limit unnecessary and counterproductive litigation costs." H.R. Rep. No. 112–98, pt. 1, at 40 (2011), 2011 U.S.C.C.A.N. 67, 69. The validity of the '844 patent is currently being litigated in *five* consolidated district court cases in the Western District of Texas ("WDTX"), and those cases are scheduled to proceed to trial two months *before* the final written decision on this petition is due.

There is no reasonable possibility of a district court stay. The district court proceedings involve *39 patents* and four additional parties, at least three of whom will litigate the validity of the '844 patent in the district court based on the same references used in this petition, regardless of whether this petition is instituted. The district court has already construed claim terms for the '844 patent, and by the time

the Board issues its institution decision (in October 2023), the parties will be deep into fact discovery. Instituting this petition will, at a minimum, generate expensive and needless duplication of effort by Patent Owner, the Board, and the district court, and it will pose the risk of inconsistent decisions.

Second, the Board should deny institution because Petitioner fails to demonstrate a reasonable likelihood that it will prevail on the merits. The '844 patent discloses a malware classifier that can turn features on or off at the classification stage, based on static data points extracted from the executable being classified. But none of the prior art on which Petitioner relies teaches or suggests those ideas. Instead, the cited art discloses *training* of malware classifiers in the standard manner—*i.e.*, identifying and selecting features *at the training stage*, and then, at the subsequent classification stage, extracting and evaluating those preselected features, irrespective of the executable being classified. Because none of the cited art suggests turning features on or off at the classification stage, let alone based on static data points of the executable that represent predefined character strings, the petition is not reasonably likely to succeed and must therefore be denied.

II. THE BOARD SHOULD EXERCISE ITS DISCRETION TO DENY INSTITUTION UNDER 35 U.S.C. § 314(a).

In view of the parallel litigations, in which a district court will adjudicate—prior to the Board—the same grounds raised in this petition, the Board should exercise its discretionary denial authority under 35 U.S.C. § 314(a).

Because Congress's goal in enacting post-grant proceedings like IPRs was “to ... limit unnecessary and counterproductive litigation costs,” Director Vidal has instructed the Board not to institute IPRs that would “increase, rather than limit, litigation costs” being incurred in parallel litigation. Director's Memo, *Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation* (“Guidance”) at 1 (June 21, 2022).

Instituting this IPR would cause just such an increase. The reasons are easy to see: a district court stay is out of the question, and the district court will consider the very same arguments presented in the petition. Accordingly, for the next year, the Board and the district court will labor to decide the very same issues with the district court well ahead of—and scheduled to reach a final decision months before—the Board. The year-long parallel tracks of two adjudicatory bodies will “increase, rather than limit, litigation costs,” create “potential conflict between the ... (PTAB) and district court proceedings,” and generally increase “the potential for inefficiency and

gamesmanship.” Guidance at 1. This is precisely the situation that Director Vidal instructed the Board to avoid when she instructed the Board to take “a holistic view of whether efficiency and integrity of the system are best served by denying or instituting review.” *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 6 (PTAB Mar. 20, 2020).

Confirming this conclusion, every one of the *Fintiv* factors weighs for discretionary denial:

(1) A stay of the co-pending district court litigation—involving 39 patents and five defendants—is vanishingly unlikely. Petitioner cannot avoid that trial because it has petitioned for IPR of only some of the patents asserted against it. And this IPR will similarly not narrow the issues on which discovery and trial must proceed—the ’844 patent is currently asserted against four other defendants, three of whom have not petitioned for IPR of the ’844 patent. Moreover, none of Petitioner’s codefendants have agreed not to litigate the validity of the ’844 patent in the district court regardless of this institution decisions by the Board. Claims have been construed, and discovery is well underway.

(2) Trial in the district court is expected two months *before* a final written decision.

(3) The parties and district court have already invested heavily in carrying

that parallel litigation past claim construction—*six months* before any anticipated institution decision—and well into discovery.

(4) The issues raised in the petition are the same as those the district court has before it, and Petitioner's narrow stipulation will not change that because the stipulation does not apply to all of Petitioner's co-defendants, who have raised the same prior art before the district court.

(5) Petitioner will get its chance to make its arguments before the district court because Petitioner is a defendant in the parallel litigation.

(6) As shown below, the petition's grounds of unpatentability are weak. Thus, every *Fintiv* factor here weighs for discretionary denial. The district court is well on its way to an efficient resolution of the disputes between Patent Owner, Petitioner, and Petitioner's codefendants. Both the goals of the AIA and the general welfare of the patent system would be best served by denying institution.

A. Factor 1: The district court has not granted and will not grant a stay.

The first *Fintiv* factor asks “whether the court granted a stay or evidence exists that one may be granted if a proceeding is instituted.” *Fintiv*, IPR2020-00019, Paper 11 at 6. This factor weighs strongly in favor of denial because the answer to both questions is **no**: no stay has been requested or ordered, and Petitioner has identified

no evidence that one will (or even might) be. To the contrary, specific evidence indicates that a stay is vanishingly unlikely.

First, the district court considers issue simplification “the most important factor” in deciding whether to stay patent litigation¹—and here, a stay would not simplify anything. As circumstances stand now, the validity of the ’844 patent will

¹ See, e.g., EX-2003, *Sonrai Memory Ltd. v. Kingston Tech. Co., Inc.*, No. 6:21-cv-1284-ADA, slip op. at *4-5 (W.D. Tex. Oct. 18, 2022) (denying a stay following IPR institution because issue simplification was the “most important factor” and the Court had already held a *Markman* hearing and “expended significant resources”); EX-2004, *mCom IP, LLC v. Cisco Sys., Inc.*, No. 6:22-00261-ADA, slip op. at *5 (W.D. Tex. Oct. 20, 2022) (finding “simplification is not likely” where the IPR did not challenge all of the claims asserted in the district court, since the other claims would still require a trial); EX-2005, *Intellectual Ventures I LLC v. Hewlett Packard Enterprise Co.*, No. 6:21-cv-00596-ADA, slip op. at *7 (W.D. Tex. Dec. 22, 2022)(denying a stay following IPR institution , as even IPR estoppel would not simplify the invalidity case, given the defendants’ reliance on prior art systems).

be litigated in the consolidated action whether this petition is instituted or not, because the patent is asserted against four other defendants, none of whom have promised not to pursue invalidity in the district court based on this institution decision.

What is more, the validity of the '844 patent will be litigated in the district court alongside issues regarding *38 other* patents asserted by and against Petitioner and *four other* defendants—issues that the district court is highly unlikely to delay deciding in favor of letting the Board proceed with *this* IPR. The case against Petitioner alone involves seventeen patents—eleven asserted by Patent Owner and six asserted in counterclaims by Petitioner.

The table below illustrates the patents, parties, and IPR petitions involved in this highly complex litigation:

**Table 1. Patent Owner's Asserted Patents
Against Each WDTX Defendant and Resulting Review Petitions**

Patent	Kaspersky	CrowdStrike	Forcepoint	Sophos	Trend Micro
8,418,250	Asserted; no IPR	IPR2023-00289	N/A	IPR2023-0699	IPR2023-00655
8,726,389	Asserted; no IPR	IPR2023-00556	Asserted; no IPR	IPR2023-00633	Asserted; no IPR
10,599,844	Asserted; no IPR	Asserted; no IPR	Asserted; no IPR	IPR2023-00528	IPR2023-00662
9,413,721	N/A	N/A	Asserted;	IPR2023-	N/A

Patent Owner's Preliminary Response
to Petition for IPR2023-00662
Patent No. 10,599,844

Patent	Kaspersky	CrowdStrike	Forcepoint	Sophos	Trend Micro
			no IPR	00491	
9,578,045	IPR2023-01199	IPR2023-00124	N/A	Asserted; no IPR	IPR2023-00677
10,257,224	IPR2023-01011	IPR2023-00126	N/A	Asserted; no IPR	IPR2023-00657
10,284,591	IPR2023-00895	IPR2022-01522	N/A	Asserted; no IPR	IPR2023-00692
8,856,505	Asserted; no IPR	IPR2023-01159	N/A	Asserted; no IPR	Asserted; no IPR
8,181,244	N/A	IPR2023-01158	N/A	Asserted; no IPR	Asserted; no IPR
8,201,243	Asserted; no IPR	IPR2023-01052	Asserted; no IPR	IPR2023-00731	Asserted; no IPR
8,719,932	Asserted; no IPR	IPR2023-01051	Asserted; no IPR	IPR2023-00732	Asserted; no IPR
11,409,869	IPR2023-01160	IPR2023-01053	Asserted; no IPR	PGR2023-00031	Asserted; no IPR
8,763,123	N/A	IPR2023-1050	N/A	Asserted; no IPR	N/A
8,438,386	N/A	N/A	IPR2023-00784	N/A	N/A
10,025,928	N/A	N/A	IPR2023-00786	N/A	N/A

As U.S. Magistrate Judge Gilliland noted during a December 9, 2022 hearing, this is “a phenomenal case that I’m hoping one day maybe people will write songs about because it is about as complicated as you’re going to get with patent cases.” (EX-2017 at 52.)

Second, the district court has been clear in public comments that it will not

stay cases where, as here, the court can resolve the case as quickly as the Board can:

[T]here's no unfairness to either side by not staying a case for the PTAB when I can try the case as quickly as they can resolve it.... And so far, I think that's proven to be true.

(EX-2008 at 3.) Indeed, as commentators have observed, "Judge Albright almost never stays proceedings to let the PTAB make its decision first." (*Id.* at 2.)

Judge Albright's public comments apply squarely to the situation here. The median time-to-trial in WDTX is 28.2 months. On that timeline, trial is expected in this case in mid-July 2024. Trial is currently scheduled for August 2024. That Judge Albright has made public comments indicating that his practice is to deny a stay in precisely the circumstances presented here is further evidence (if any were needed) that he would not stay the WDTX case in response to institution of this petition.

Given Judge Albright's stated policies and the complexity of the parallel proceedings, no "speculat[ion]" is necessary. To the contrary, imagining circumstances in which Judge Albright would cast aside his stated preferences and policies to grant the extraordinary remedy of a stay of *this* parallel case in particular requires far-reaching speculation.

Even where no stay has been requested, the Board has found Factor 1 to favor discretionary denial where (as here) "Petitioner present[ed] no evidence that it has

requested a stay [] or that a stay is likely.” *Luxshare Precision Industry v. Amphenol Corp.*, IPR2022-00132, Paper 10, at 9 (PTAB May 3, 2022). And such a decision is particularly appropriate where the circumstances show that “a stay ... is unlikely.” *Nintendo Co. v. Gamevice, Inc.*, IPR2020-01197, Paper 13, at 11-12 (PTAB Jan. 12, 2021); *Samsung Elecs. Co. v. Clear Imaging Research LLC*, IPR2020-01399, Paper 13 at 11-12 (PTAB Feb. 3, 2021) (holding Factor 1 may weigh for denial where there is evidence that the district court is unwilling “to avoid duplicative efforts and await the PTAB’s final resolution”); *Motorola Mobility LLC v. Ironworks Patents, LLC*, IPR2021-00421, Paper 11 at 7-8 (PTAB July 22, 2021).

Given the full circumstances present here, it is not plausible that the district court will stay its multi-patent, multi-defendant case so that the Board can reach a *slower* decision on *just one* patent. Factor 1 supports discretionary denial. *Samsung*, IPR2020-01399, Paper 13 at 11-12; *Motorola*, IPR 2021-00421, Paper 11 at 7-8.

B. Factor 2: A district court trial is expected near the projected statutory deadline.

Factor 2, which considers the implications of the district court and IPR schedules, also weighs for denial. “[A] parallel proceeding in an advanced state implicates considerations of efficiency and fairness, which can serve as an independent reason to apply discretion to deny institution.” *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019,

Paper 15 at 11-12, 17 (PTAB May 13, 2020) (informative); Guidance at 1.

Here, Institution here would merely serve to “increase, rather than limit, litigation costs.” Guidance at 1. Specifically, it would double the parties’ costs and obligations to litigate the ’844 patent’s validity in separate forums. The IPR trial would overlap almost entirely with the parallel district court litigation. WDTX’s median time-to-trial is 28.2 months (EX-2010 at 37), indicating a trial by mid-July 2024—a date consistent with the currently scheduled trial date of August 19, 2024. (EX-2009 at 9.) Should the Board institute, its final written decision deadline would be October 17, 2024, two months after the scheduled trial date. In other words, the IPR proceedings will overlap almost entirely with the parallel litigation and likely will not produce a result before the trial.

But the likelihood that the Board’s decision will not come until months after the district court’s is only part of the story. The Board should consider “the duplicative efforts that will occur during the [] duration of the inter partes review.” *NXP USA, Inc. v. Impinj, Inc.*, PGR2022-00005, Paper 20 at 7 (PTAB Aug. 25, 2022). And here, even in the unlikely event that the Board reaches decision several months before the deadline, the parties will spend nearly a year expending precisely the “duplicative,” “unnecessary[,] and counterproductive litigation costs” that the AIA was designed to avoid. *Fintiv*, IPR2020-00019, Paper 11 at 5 (citing *Magellan Midstream*

Partners L.P v. Sunoco Partners Mktg. & Terms. L.P., IPR2019-01445, Paper 12 at 10 (Jan. 22, 2020)); Guidance at 1.

Because the district court will not stay the litigation, and the district court will resolve validity before the Board reaches its final written decision, the second *Fintiv* factor also favors discretionary denial.

C. Factor 3: The parties will have invested significant resources in the district court proceeding.

Factor 3—the resources invested in the parallel district court proceeding—strongly favors denial.

The parties have invested significant resources in the parallel litigation over the last 15 months. For example, the parties long ago exchanged infringement and invalidity contentions (in July 2022 and September 2022 respectively), and fact discovery is underway. Claim construction, too, is long since complete (EX-2007), “indicat[ing] that the court and parties have invested sufficient time in the parallel proceeding to favor denial.” *Fintiv*, IPR2020-00019, Paper 11 at 10.

And while *Fintiv* directs consideration of resource investment at the time of institution, the overlapping schedules here counsel the Board to further consider the redundant resources that *will be* wasted should IPR be instituted. *Id.* at 6 (requiring “a holistic view of whether efficiency and integrity of the system are best served by

denying or instituting review”).

The investment in this case is colossal. The district court's *Markman* order that included the '844 patent was 20 pages, without even considering the Court's substantive analysis or the additional *Markman* orders issued in this case. (EX-2009.) The Court considered and construed five claim terms of the '844 patent alone. (EX-2007.) And as of the date of Patent Owner's preliminary response, the parties have engaged in extensive discovery on the parties' contentions. The Court has held multiple discovery hearings. Furthermore, Patent Owner's experts have already invested an extensive number of hours preparing for their reports by reviewing the patent, prior art, the defendants' invalidity contentions, and source code. *See Regeneron Pharms. Inc. v. Novartis Pharma AG*, IPR2020-01317, Paper 15 at 17 (PTAB Jan. 15, 2021) (denying institution where the parties had expended considerable resources in claim construction, substantial fact discovery, and preparation for expert reports).

Petitioner's delay in filing this petition further weighs for denial. Petitioner waited nearly 12 months after receiving Patent Owner's March 4, 2022 complaint, eight months after receiving Patent Owner's infringement contentions, and six months after serving its invalidity contentions before filing its petition. (EX-2014 at 1-2.) Petitioner offers no explanation for its delay, which weighs for denial. *Fintiv*,

IPR2020-00019, Paper 11 at 11-12; *Ericsson Inc. v. Collision Comms., Inc.*, IPR2022-01233, Paper 12 at 15 (PTAB Jan. 19, 2023) (finding that where the parties invested significant resources and “Petitioner waited almost a year” to file its petition, “this factor weighs decidedly in favor of discretionary denial”); *Regeneron*, IPR2020-01317, Paper 15 at 17 (finding that even where the petition was filed “less than a month after learning of the complaints,” the investment by the parties in the parallel proceeding “outweighs the effort expended so far in this proceeding”).

D. Factor 4: The issues and parties overlap significantly.

Factor 4—the degree of overlap between the issues in the petition and those before the district court—also supports denial.

Regarding this fourth factor, *Fintiv* explains that “concerns of inefficiency and the possibility of conflicting decisions [are] particularly strong” where “the Board [is] presented with substantially identical prior art arguments ... at issue in district court.” *Fintiv*, IPR2020-00019, Paper 11 at 12. “Accordingly, if the petition includes the same or substantially the same claims, grounds, arguments, and evidence as presented in the parallel proceeding, this fact has favored denial.” *Id.*

Here, the petition identifies the same prior art that Petitioner and its co-defendants have put before the district court (*see, e.g.*, EX-2019, CrowdStrike Invalidity Contentions at 12; EX-2012, ForcePoint Invalidity Contentions at 61; EX-2013,

Sophos Invalidity Contentions at 14). Indeed, Petitioner admits that “Petitioner has identified the Tahan, Menahem, and Schultz prior art in Defendant’s invalidity contentions filed in the parallel litigation.” Petition 69.(Petitioner at 69.) Neither Petitioner nor any of its co-defendant has offered any type of stipulation that, if the Board institutes review here, they will not pursue the same grounds or references before the district court.² Consequently, this fact strongly favors denial.

E. Factor 5: Both parties are parties in the parallel litigation.

Factor 5—which considers overlap of the parties between the petition and the parallel litigation—favors denial because (1) Petitioner is a defendant in the parallel litigation and, as such, will have a full, fair opportunity to litigate the petitioned grounds, and (2) the issues Petitioner raises are the same ones its co-defendants have *also* raised in the parallel litigation.

Under *Fintiv*, “[i]f a petitioner is unrelated to a defendant in an earlier court proceeding, the Board has weighed this fact against exercising discretion to deny

² Thus far, only one of those co-Defendants (Sophos) has offered any sort of stipulation, and that stipulation hinges on whether the Board institutes trial in a different IPR (IPR2023-00528), that raises different art.

institution.” *Fintiv*, IPR2020-00019, Paper 11 at 13-14. Here, by contrast, Petitioner is not just related, it is itself a defendant in the litigation. Accordingly, the circumstances raise no fairness concerns that denial would deprive Petitioner of its “day in court.” See *Vizio, Inc. v. Maxell, Ltd.*, IPR2022-01459, Paper 7 at 21-22 (PTAB Apr. 11, 2023) (“Because Petitioner is the litigation defendant, factor 5 weighs in favor of discretionary denial”); *Nokia of Am. Corp. v. Godo Kaisha IP Bridge*, IPR2022-00755, Paper 9 at 13 (PTAB Nov. 2, 2022) (holding that Factor 5 weighed for discretionary denial where Petitioner was defendant in parallel proceeding).

That other defendants are *also* litigating the petitioned issues at the district court further suggests denial. *Fintiv* holds that, “[e]ven when a petitioner is unrelated to a defendant, however, if the issues are the same as, or substantially similar to, those ... about to be litigated, or other circumstances weigh against redoing the work of another tribunal, the Board may, nonetheless, exercise the authority to deny institution.” *Fintiv*, IPR2020-00019, Paper 11 at 14. Because of this consideration, *Fintiv* goes on to require that “[a]n unrelated petitioner ... discuss why addressing the same or substantially the same issues would not be duplicative of the [other] case even if the petition is brought by a different party.” *Id.*

Petitioner offers no such explanation here. That other defendants are currently litigating the same issues raised in the petition raises serious concerns of duplicative

work by adjudicators, and any public-policy concerns regarding patent validity will be thoroughly addressed by the parallel litigation. Thus, that *other* defendants in the district court proceedings are *also* litigating the petitioned grounds weighs yet further for discretionary denial.

F. Factor 6: The petition does not present “compelling evidence of unpatentability.”

Factor 6—the merits of the Petition—also counsels for denial because, as detailed below, the petition fails to show a “reasonable likelihood” that Petitioner would prevail on any challenged claim. *See* 35 U.S.C. § 314(a). For example, none of the cited art teaches, suggests, or renders obvious the idea of a machine-learning malware classifier that turns features on or off at the classification stage (let alone based on the values of static data points extracted from the executable and representing predefined character strings). Because the petition fails to show a reasonable likelihood of success on the merits, the petition may not be instituted.

Even if the Petition did have a “reasonable likelihood” of succeeding (it does not), it would still not be a “compelling” challenge that precludes discretionary denial. Guidance at 4. “[T]he compelling merits standard is a higher standard than the standard for institution set by statute.” *Commscope Techs. v. Dali Wireless, Inc.*, IPR 2022-01242, Paper 23, at 3 (PTAB Feb. 27, 2023). That higher standard requires

that the Petition is “highly likely” to succeed. *Id.* at 4. The opposite is true here.

* * *

In sum, all six factors weigh for discretionary denial. Accordingly, Patent Owner respectfully requests that the Board exercise its discretion to deny institution.

III. THE PETITION IS NOT REASONABLY LIKELY TO SUCCEED.

The Board should deny institution because the petition fails to show a reasonable likelihood of prevailing on any challenged claim. The petition presents two grounds of obviousness:

Ground 1	Section 103(a)	Rendered obvious by the combination of Tahan and Menahem	Claims 1-5, 8, 10-14, and 17
Ground 2	Section 103(a)	Rendered obvious by the combination of Tahan, Menahem, and Schultz	Dependent Claims 6, 7, 9, 15, 16, 18, and 20

Ground 1 is that the combination of Tahan and Menahem purportedly rendered independent claims 1, 10, and 17 and dependent claims 2-5, 8, and 11-14 obvious. Ground 2 is that the combination of Tahan, Menahem, and Schultz rendered dependent claims 6, 7, 9, 15, 16, 18, and 20 obvious. Neither ground has a reasonable likelihood of succeeding.

Ground 1 fails on the independent claims (1, 10, and 17):

- *First*, the cited combination fails to suggest a machine learning malware

classifier wherein “features of the feature vector are selectively turned on or off” *at the classification stage*, let alone based on the “one or more static data points” of the file being classified. Instead, the cited combination teaches only the traditional idea of selecting features at the earlier *training* stage, before the system ever sees the file being classified.

- *Second*, Petitioner fails to explain how or why the cited references would be combined to perform feature selection based on extracted values that represent “predefined character strings.” Though Menahem may suggest using static data points that represent predefined character strings (*e.g.*, PE file header values), the petition fails to explain how that idea could be used to modify Tahan’s system, let alone why a POSITA would have been motivated to make that modification.

Because Ground 1 fails for independent claims 1, 10, and 17, it also fails for dependent claims 2-5, 8, and 9-14.

Ground 2 fails for similar reasons as Ground 1 because it adds Schultz only to purportedly teach the limitations *added* by dependent claims 6, 7, 9, 15, 16, 18, and 20. (Petition at 55-69.) Because Ground 2 includes no argument that Schultz remedies any of Ground 1’s shortcomings respecting the independent claims, and the claims it challenges all depend on those independent claims, Ground 2 cannot render

any of the challenged claims obvious.³

A. Background

1. The state of the art

Every day, thousands of new executable files (“executables”) are created and distributed across computer networks. (EX-1001 at 1:7-8.) It is the job of computer security software to quickly recognize which of those executables is malicious or otherwise unwanted, and to prevent execution of them. (*Id.* at 1:8-15.)

One conventional approach is to perform “static analysis,” whereby the security software analyzes the executable’s code to identify known malicious code blocks within the executable. (*Id.* at 2:15-17.) That conventional approach had two drawbacks: (1) it could not identify malicious files that it had not seen before, and (2) it was slow at evaluating encrypted files because it required the security software to first decrypt and disassemble the code.

³ Because this is a preliminary response, Open Text does not address every limitation of every claim. But this is not an admission that any limitation not discussed is disclosed or rendered obvious by the cited references. Open Text reserves the right to argue that they are not should the Board institute trial.

One solution to the first problem (*i.e.*, identifying malicious code that has not been seen before) is to use a “machine learning classifier.” (*Id.* at 1:20-40.) Machine learning classifiers are software programs that are “trained” to classify new, unknown executables by studying a large group of *known* executables (*e.g.*, “known malicious executable files, known benign executable files, and [known] potentially unwanted executable files”). (*Id.*)

During a “training stage,” such systems analyze a large group of known executables to (1) determine a set of “features” of the executables that can be used to accurately classify the executables as malicious or benign, and (2) build a classifier that can classify a new executable based on this set of features. The “trained” classifier is a predictive model that accepts a set of features as input and outputs a predicted classification.

During a “classification stage,” the system receives a previously unseen executable, extracts from it values for each of the features that were identified at the training stage, inputs those values for the features into the trained classifier, and thus obtains a predicted classification for the executable. (EX-2001 at ¶25.)

Because the computational complexity (and therefore the required computational time and resources) grows with the number of features a classifier has been trained to consider, the *training stage* sometimes includes a process known as

“feature selection” or “feature pruning”—*i.e.*, the process of identifying the features that are most effective and/or computationally inexpensive and configuring the classifier to consider only those features. Feature selection at the training stage reduces the computational burden at the classification stage because the classifier will have to extract fewer features and/or perform less complex mathematical computations. (Ex-2001 at ¶25.)

2. The '844 Patent

The '844 patent improves upon the prior art by introducing a technique of performing feature selection at the classification stage (rather than merely during the training stage). Performing feature selection at the classification stage allows the system to dynamically calibrate its performance based on the content of the executable because the system can turn “off” features that are unlikely to be valuable for classification of a given executable. Because features that are turned “off” need not be considered (or even extracted) by the classifier, this technique accelerates the performance of the classification stage by relieving the system from having to determine values for and consider features that are unlikely to help the classifier get an accurate and fast result. The technique thus improves upon prior art “static” evaluation, whereby the classifier had to always evaluate, or “test,” every feature that was predetermined during the training stage.

In the “training stage,” the system described in the ’844 patent examines a training set of executables known to be malicious, benign, or potentially unwanted. (EX-1001 at 4:22-67.) The system extracts “static data points” from the executables, such as “header information, section information, import and export information, certificate information, resource information, string and flag information, legal information, comments, and/or program information (*e.g.*, APIs and DLLs), among other examples.” (*Id.* at 5:1-6:58.)

The specification provides many exemplary embodiments of static data points (*Id.* at Table 1.1, 8:53-56), and a POSITA would recognize that many of these embodiments correspond to fields defined by the “Portable Executable” file format, such as “file size,” “PE header time,” and “linker version.” (*Id.* at 3:35-46, 4:22-24 (describing extraction of static data points from PE files).) In such embodiments, a static data point may include a field name (*i.e.*, a predefined character string such as “File Size”) and a field value (*e.g.*, a number such as 4982784). (*Id.* at 8:53.)

The system then uses the static data points to generate feature vectors. (*Id.* at 8:22-9:40) (describing conversion of static data points to a feature vector, though in the context of the classification stage.) The specification discloses embodiments wherein the vector includes one or more features for each static data point. (*See id.* at 8:49-9:40.)

Finally, the system uses the feature vectors to train a classifier. (EX-1001 at 8:14-21.) The specification discloses that the classifier may be of various types, such as an “artificial neural network (ANN), a decision tree, association rules, inductive logic, a support vector machine, clustering analysis, and Bayesian networks, among other examples.” (*Id.*)

In the “classification stage,” the system of the ’844 patent uses the classifier to classify a new executable. It receives the new executable for classification and extracts the static data points from it. (*Id.* at 12:4-10.)

The system then generates a feature vector from the static data points as before. (*Id.* at 12:48-50.) Unlike prior art systems, however, “features of the feature vector may be selectively ***turned on or off*** based on evaluation of whether a value of a static data point [from the file being classified] is within a predetermined range.” (*Id.* at 12:60-64 (emphasis added); *see also id.* at 7:57-8:22.) For example, “where a static data point extracted for evaluation is a file size (*e.g.*, numeric value), the learning classifier may turn on/off features for evaluation based on whether [] the file size of the file is within a certain range.” (*Id.* at 7:62-64.) Thus, at the classification stage, the classifier would consider certain features only if the executable being classified meets certain conditions, such as the size of the executable being above a given threshold, the executable including a certain type of content (*e.g.*, legal copyright

information), or the executable including a particular permutation of that content's subtypes (e.g., for legal information, "timestamp data," "licensing information," "copyright information"). (EX-1001 at 7:57-8:22.)

In short, the system decides which features to turn on or off at the classification stage based on the values of the static data points extracted from the executable being classified at that stage. The set of features ultimately extracted from the executable being classified may be used to create one or more feature vectors that are input into the appropriate classifier. (*Id.* at 8:49-9:40.)

Claim 1 recites a method that corresponds to the classification stage of the invented technique:

A computer-implemented method comprising:
extracting a plurality of static data points from an executable file without decrypting or unpacking the executable file, wherein the plurality of static data points represent predefined character strings in the executable file;
generating a feature vector from the plurality of static data points using a classifier trained to classify the plurality of static data points based on a collection of data comprising known malicious executable files, known benign executable files, and known unwanted executable files,
wherein the collection of data comprises at least a portion of the plurality of static data points, and
wherein one or more features of the feature vector are selectively turned on or

off based on whether a value of one or more static data points from the plurality of extracted static data points is within a predetermined range; and evaluating the feature vector using support vector processing to determine whether the executable file is harmful.

(*Id.* at claim 1.)

The claimed method begins with extracting, from the executable being classified, “static data points [that] represent predefined character strings.” For example, a static data point may correspond to the size of the executable, as represented in the executable’s PE header by a “predefined character string” (*e.g.*, a “File Size”) and a corresponding numerical file size (*e.g.*, 2MB).

During prosecution of the ’844 patent, the applicants distinguished these steps from prior art that started by simply “split[ting] [the executable] into non-overlapping file chunks of fixed length” rather than purposefully extracting static data points that “represent *predefined* character strings.” (Ex. 1007 at 156 (emphasis added).) There, the applicants explained that “fixed-length file chunks do not represent predefined character strings” because the chunks are merely “random (*e.g.*, undefined) character string[s]” having no significance at the time they are extracted. (*Id.*)

Once the static data points are extracted from the executable, a feature vector representing the executable is created by a machine learning “classifier” that was previously “*trained*” (past tense) on “known malicious,” “known benign,” and

“known unwanted.” Though the classifier was already trained, “one or more features of the feature vector are selectively turned on or off based on whether a value of [the] ... static data points ... is within a predetermined range.” Thus, for example, the classifier might turn certain features “off” if the static data point representing the file size of the executable has a value lower than some predefined threshold (*e.g.*, 5MB).

Finally, the classifier classifies the executable by analyzing the generated feature vector. The analysis uses support vector processing.

Claim 17 recites a system that performs a variation of the method recited in claim 1: the system of claim 17 extracts predefined character strings (rather than static data points), uses those to create a feature vector, and turns certain features on or off based on the values of those predefined character strings.

3. Claim construction

The parties agree that no claim terms need be construed for this petition, and all claim terms should carry their plain and ordinary meaning in light of the intrinsic record. (Petition at 9-10.)

As referenced above, the parties are involved in parallel litigation where the district court has issued an interlocutory Claim Construction Order rejecting various proposals advanced by Petitioner and other defendants. (EX-2007 at 17-20.) The

Board's regulations require that the Board consider those constructions. 37 C.F.R. § 42.100(b).

- **“executable file.”** The district court held that the phrase “executable file” carries its plain and ordinary meaning. (*Id.* at 17.) The court rejected a proposal that would have required “executable files” to be “encrypted or packed.” (*Id.*)
- **“character strings” being “within a predetermined range.”** The district court held that it was not indefinite for a “character string” to be “within a predetermined range,” as recited in claim 17, and that this phrase should be given its plain and ordinary meaning. (*Id.* at 18.)
- **“within a predetermined range.”** The district court held that the phrase “within a predetermined range” carries its plain and ordinary meaning, and it defined “range” as a “numerical range or a subset of a ‘particular scale’ with at least a lower limit or an upper limit, but not necessarily both.” (*Id.*) The district court rejected a proposal that a value “within a predetermined range” must be “between a predefined lower and upper limit.” (*Id.*)
- **“support vector processing.”** The district court held that “support vector processing” carries its plain and ordinary meaning, and that this meaning is “processing that involves a support vector.” (*Id.*) The court rejected the

argument that “support vector processing” is limited to “processing with a support vector machine.” (*Id.*)

- **“at least one processor.”** The district court held that this phrase has its plain and ordinary meaning, and it rejected the argument that the phrase is a means-plus-function limitation under § 112(f). (*Id.* at 15.)

4. The level of ordinary skill

For purposes of this proceeding, Patent Owner does not dispute Petitioner's articulation of the level of ordinary skill. (Petition at 9.)

B. Overview of the cited art.

1. Overview of Tahan

Tahan is an academic paper that discusses various machine learning approaches for detecting malware in executables. (EX-1003 at 1.)

Tahan discloses three different approaches to malware classification. *First*, Tahan's introduction describes the state of the prior art in machine learning (ML) malware classifiers. (EX-1003 at 1-5.) *Second*, Tahan proposes a new system called “Mal-ID basic.” (EX-1003 at 5-11.) *Third*, it discloses a variation called “Mal-ID with ML Generated Models.” (EX-1003 at 11-13.)

i. Tahan's Introduction

Tahan begins with an “Introduction” (EX-1003 at 1-5), in which it discusses

the general idea of “using machine learning (ML) techniques to produce a binary classifier that is able to distinguish malware from benign files.” (EX-1003 at 2.)

The Introduction's description of that general idea matches the description of the prior art “training stage” addressed above. *See supra* Section III.A.1 (“State of the Art”). Specifically, Tahan describes a process in which: (1) features are extracted from executables in a training set, (2) features are selected, such that “redundant and irrelevant features are discarded,” and (3) “[t]he last phase is creating a classifier using the reduced feature vectors created in phase 2.” (EX-1003 at 2.) In other words, the Introduction describes feature extraction and selection at the training stage. (EX-2001 at ¶49.)

The Introduction does not describe the classification stage in any meaningful detail. It simply notes that the classifier created during the training stage can be of any type, and it surveys the efforts of other researchers to use different types of features and classifiers. (*Id.*)

Thus, Tahan's Introduction merely provides a background on feature extraction, feature selection, and classifier training at the training stage—the same process described above as prior art. *See supra* Section III.A.1 (“State of the Art”).

ii. Mal-ID Basic

Following the Introduction, Tahan introduces a *new* machine learning system

called “Mal-ID.” (EX-1003 at 5-13.) That system is fundamentally different from those described in the Introduction.

Tahan teaches that the “basic insight” that motivated Mal-ID is that “both legitimate developers and hackers” use the same “higher level development platforms” and “the same specialized libraries.” (EX-1003 at 5.) Consequently, the “main idea” of Mal-ID is “to detect the file segments that originated from the development platform or from a benign third-party library (termed here the Common Function) and then disregard those segments” while “the remaining segments would be compared to determine their degree of resemblance to a collection of known malware.” (EX-1003 at 7.) Tahan calls this process “common segment analysis.” (*Id.*)

The first variation of Mal-ID that Tahan discloses is called “Mal-ID basic.” (EX-1003 at 9-11.)

Mal-ID basic proceeds in two phases: (1) a “setup phase,” in which data is gathered from a training set of executables, and (2) a “detection phase,” in which “a previously unseen” executable is analyzed and classified using the data gathered in the setup phase. (EX-1003 at 7.)

Setup Phase

In the setup phase, a training set of known executables is analyzed to determine (1) the Common Functions (*i.e.*, those found in benign executables) and (2) the

“Threat Functions” (*i.e.*, those found in malicious executables). (*Id.*)

The system starts by “collecting two kinds of files: benign and malware.” (EX-1003 at 7.) Then, a “CFL repository [containing the Common Function segments] is constructed from benign files and [a] TFL repository [containing the Threat Function segments] is constructed from malware files.” (*Id.*) “The CFL is constructed first and then the TFL.” (EX-1003 at 9.)

To construct the CFL, each benign “file is divided into 64 equal length segments” (*id.*), which are each further “broken into 3-grams (three consecutive bytes)” (EX-1003 at 7.) Moreover, “[t]he training files are randomly divided into 64 groups.” (*Id.*)

Then, for every combination of 3-gram and file group, the system records in the CFL (1) whether the 3-gram appears in the files of that group (“3-gram-files-association”) and (2) which segment of the file the 3 gram appears in (“3-gram-relative-position-within-file”). (EX-1003 at 8-9.)

The TFL is then constructed: “Each file from the malware collection is broken into ... 64-byte segments,” “[e]ach segment is broken into 3-grams and then tested against the CFL” such that “[s]egments that are not in the CFL are added to the TFL.” (EX-1003 at 9.) “[T]he end result is the TFL, a repository made of segments found only in malware and not in benign files.” (*Id.*)

Detection Phase

In the detection phase, the data collected in the setup phase is analyzed to classify a previously unseen executable. (EX-1003 at 9-10.)

At a high level, the classification algorithm simply (1) “[d]ivide[s] file F into S segments of length L” and (2) counts how many of those segments meet a given set of criteria. (EX-1003 at 10.) That set of criteria is:

- Unique Segment. “[A]ll duplicated segments are removed.” (*Id.*)
- Entropy Is Within Range. “[I]f the entropy is smaller than EntropyLow threshold or entropy is larger than Entropy-High ... then discard the segment.” (*Id.*) “Entropy” refers to “the entropy measure of the bytes within a specific segment candidate.” (EX-1003 at 9.)
- Not in CFL. “[I]f the CLF-MFG [*i.e.*, maximum total number of file-groups in CFL that contain the segment] is larger than zero, then discard the segment.” (EX-1003 at 10.)
- In TFL. “[I]f the TFL-MFG index [*i.e.*, maximum total number of file groups in TFL that contain the segment] is smaller or equal to the ThreatThreshold ... then discard the segment.” (*Id.*) For example, “only segments that appear two times or more [in the TFL] remain.” (*Id.*)

- TFL-Spread Within Range. “If the TFL-Spread index equals zero or if it is larger than what we term SR threshold, then discard the segment.” (*Id.*) The TFL “spread” “represents the spread of the signature’s 3-grams along the various areas for all the files in [the TFL].” (EX-1003 at 9.)
- Second Level Index Aggregation. The segment must be “found in malware and not in the CFL.” (EX-1003 at 11.)

To decide whether the executable is malicious or benign, the system simply counts how many of the executable’s segments meet the above set of criteria. (EX-1003 at 11.) “If there are at least X segments” that meet the above set of criteria, the system determines that “the file is malware,” and “otherwise, [it] consider[s] the file as benign.” (*Id.*)

Tahan describes the executable’s *3-grams* as the extracted “features,” and the “entropy,” “MFG,” and “Spread” measures calculated based on those 3-grams as “meta-features.” (EX-1003 at 9 (“Note that the features, as described above, are in fact *meta-features* as they are used to represent *features of features* (features of the basic 3-grams).”) (emphasis in original).)

As this description shows, the “feature selection” described in Tahan’s Introduction is fundamentally different from the process performed by Mal-ID. The

feature selection during the training phase of the Introduction determines *which* features will be extracted at the classification stage, whereas the setup phase of Mal-ID determines parameters that will be used to *analyze* the features (*i.e.*, 3-grams) that will be extracted at the later “detection phase.” In other words, the features that Mal-ID extracts at the “classification stage” (*i.e.*, the “detection phase”) are always the same: “S segments of length L,” regardless of what was done at the “training stage” (*i.e.*, the “setup phase”). (EX-1003 at 10.) The setup phase of Mal-ID thus does not perform “feature selection,” as described in the Introduction. (EX-2001 at ¶65.)

Moreover, this description also shows that the segments that each file is subdivided into are not “predefined character strings,” and are identical to the random, fixed-length file chunks used in wavelet decomposition, a method that the applicants explicitly distinguished their invention from, with the assent of the Patent Office. (EX-2001 at ¶66.)

iii. Mal-ID with ML

In Section 2.4, Tahan suggests “improv[ing] the Mal-ID basic method by using Mal-ID features with various classifiers, but instead of using the Mal-ID decision model ... we let various ML algorithms build the model.” (EX-1003 at 11.) This is the third different system from Tahan that Petitioner relies on in the petition.

Setup Phase

The setup phase of Mal-ID with ML starts like the setup phase of Mal-ID basic: by building the CFL and TFL in the same manner as did Mal-ID basic. (*Id.*) (“obtain a collection of segments for both the CFL and the TFL as explained in Section 2”).

The setup phase of Mal-ID with ML differs from that of Mal-ID basic in that Mal-ID with ML goes on to (1) create a fixed-length feature vector to represent each training file and (2) use the collection of feature vectors to train a machine learning classifier. (EX-1003 at 11-13.)

Mal-ID with ML creates the feature vectors as follows:

- Calculate the meta-features for each segment of each file. (EX-1003 at 11 (“For each file’s segment, we calculate the CFL-MFG, TFL-MFG and the TFL-spread based on the CFL and TFL,” and “[t]he measure is calculated as well.”).)
- Discretize the values of the meta-features by grouping them into a fixed number of bins. (*Id.* (“We discretize the numeric domain of the above features” into “sub-domains (bins)”.)

- For each file, create a feature vector indicating the number of the file's segments associated with each bin. (*Id.* (“For each file, count the number of segments associated with each bin”).)

The result is that each executable in the training set is represented by a respective feature vector with the same number of elements (*i.e.*, the number of meta-feature/bin combinations).

Mal-ID with ML then uses the extracted feature vectors to train a machine learning classifier. (EX-1003 at 12 (“An induction algorithm is trained over the training set to generate a classifier.”).) Tahan discloses that different types of classifiers can be used, and reports that the authors investigated using three particular types of classifiers: a single decision tree (“C4.5”), a Rotation Forest (“RF”), and a neural network (“NN”).

Detection Phase

At the detection phase (*i.e.*, the classification stage), Mal-ID with ML receives a new executable to classify, extracts from the executable the same set of features that were extracted from the training set and used to train the classifier at the setup phase, and inputs that feature vector into the trained classifier for classification. (EX-1003 at 13..)

Mal-ID with ML thus operates like the traditional ML methods described in

the Introduction: (1) a *training* stage selects the set of features and trains a classifier on that set of features, and (2) a subsequent *classification* stage extracts that predetermined set of features from the executable being classified and inputs those features into the trained classifier for classification. (EX-2001 at ¶73.)

2. Overview of Menahem

Menahem discloses ideas for combining multiple malware classifiers into an “ensemble.” (EX-1004 at 1.)

Like Tahan, Menahem starts with an “Introduction” describing the general process of “detecting malicious executables using Machine Learning (ML) techniques.” (*Id.*) As described above, that process involves (1) a training stage in which “[m]alicious and benign files are represented by a vector of features” that “are used to train a classifier,” and (2) a classification stage (which Menahem calls the “detection phase”) in which “an unknown file can be classified” using the trained classifier. (*Id.*) Menahem teaches that some classifiers can use features “extracted from the PE header” of executables. (*Id.*)

Menahem observes that, because different classifiers are built using different “inducer algorithms” and different “types of features,” “each classifier will perform differently,” such that some classifiers may be better or worse at classifying certain types of malware. (EX-1004 at 2.)

Menahem suggests that this problem can be solved by combining multiple types of classifiers into “multi-inducer ensembles.” (*Id.*) “The main idea of an ensemble methodology is to combine a set of classifiers in order to obtain a better composite global classifier.” (*Id.*) For example, Menahem discloses a “Majority Voting” scheme in which a group of different classifiers “vote” and the classification chosen is “the class that obtains the highest number of votes.” (EX-1004-3.) Menahem also proposes more complex ways of combining the classifiers, including techniques called “stacking” and “Troika.” (EX-1004 at 4; EX-2001 at ¶77.)

Menahem teaches that, “to make the ensemble more effective, there should be some sort of diversity between the classifiers,” and that this “diversity is obtained by using different types of inducers.” (EX-1004 at 3.)

C. Ground 1: The combination of Tahan and Menahem did not render claims 1-5, 8, 10-14, and 17 obvious.

“In an inter partes review, the burden of persuasion is on the petitioner.... Failure to prove the matter as required by the applicable standard means that the party with the burden of persuasion loses on that point—thus, if the fact trier of the issue is left uncertain, the party with the burden loses.” *Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378-79 (Fed. Cir. 2015); *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016) (noting petitioners have the

“burden from the onset to show with particularity why the patent it challenges is unpatentable”). Accordingly, a petition must include “a detailed explanation of the significance of the evidence.” 37 C.F.R. § 42.22(a)(2); *see also* 35 U.S.C. § 312(a)(3); *Lumenis Be Ltd. v. BTL Healthcare Techs. A.S.*, IPR2021-01280, Paper 8 at 22 (PTAB Mar. 9, 2022) (finding petitioner had not met its burden and declining to institute because petitioner’s argument lacked explanation and the Board was unwilling to speculate).

Here, Petitioner fails to establish a reasonable likelihood that the combination of Tahan and Menahem renders claims 1-5, 8, 10-14, and 17 obvious for at least two reasons: (1) the cited combination fails to suggest a malware classifier wherein “features of the feature vector are selectively turned on or off” *at the classification stage* (when the file is being classified) rather than at the earlier *training* stage (before the system ever sees the file being classified), and (2) the Petitioner fails to explain how or why the combination would select features (at any stage) based on “predefined character strings.” (EX-2001, at ¶¶80-82.)

1. The cited art fails to suggest turning features on or off at the classification stage.

An important distinction between the machine learning classifiers claimed and those disclosed in the cited art is that the claimed classifiers can turn features on or

off at the *classification* stage rather than only during the training stage.

The claims recite the steps of the classification stage, wherein a previously unseen executable is analyzed and classified. For example, claim 1 recites:

- (1) “extracting ... static data points from an executable,”
- (2) “generating a feature vector from the ... static data points using a classifier *trained* to classify the ... static data points,” and
- (3) “evaluating the feature vector ... to *determine whether the executable ... is harmful*” (*i.e.*, classifying the executable).

(EX-1001 at Claim 1 (emphasis added).)

Importantly, the “static data points” are extracted—and the “feature vector” therefore derived—from *the executable that is being classified* (*i.e.*, from the executable that is “evaluat[ed] ... to determine whether the executable is harmful”). That executable is not even known until the classification stage. Therefore, claim 1 does not recite the steps of the earlier training stage; it necessarily recites the steps of the *classification* stage.

Because the step of “generating a feature vector” happens at the classification stage, the requirement that this step include turning features “on or off” must likewise happen at the classification stage. Indeed, the features must be turned on or off “based on ... [the] ... static data points” extracted from the executable being

classified, which again, is not known until the classification stage. Claim 1 thus requires that the system turn features “on or off” *at the classification stage*.

The benefit of turning features on or off at the classification stage is that classifier performance can be optimized *based on the executable being classified*. Unlike the prior art, the classifier is not locked into evaluating the full set of features that was predetermined at the training stage, without regard to the executable being classified at the classification stage. (EX-2001, at ¶¶31-32.)

Neither Tahan nor Menahem (alone or in combination) suggests the idea of turning features on or off at the classification stage. Instead, each teaches only the traditional idea of performing feature selection at the training stage.

The petition's contrary arguments (Petition at 20-22) indiscriminately coningle elements of four different systems discussed in the cited references: (1) the traditional ML systems disclosed in Tahan's Introduction, (2) Tahan's Mal-ID Basic, (3) Tahan's Mal-ID with ML, or (4) Menahem's ensemble techniques. The petition conflates elements from those four systems without ever acknowledging that the elements come from different systems, explaining how the elements could be combined to form the recited system, or explaining why a POSITA would have been motivated to do so. (*See* Petition at 20-22).

None of the four systems addressed in the petition suggests turning features

on or off at the classification stage, and thus, their combination does not either.

i. Tahan's Introduction

Tahan's Introduction does not describe an ML malware classifier that turns features on or off at the classification stage. Instead, as shown above (*supra* Section III.B.1.i), Tahan's Introduction simply describes the traditional ML technique of performing feature selection *at the training stage*: (1) features are extracted from executables in a training set, (2) "feature selection" occurs, where "redundant and irrelevant features are discarded," and (3) "[t]he last phase is creating a classifier using the reduced feature vectors created in phase 2." (EX-1003 at 2-3.) Creating the classifier is the last step in the training stage. Thus, the steps disclosed in the Introduction describe "feature selection" during the training stage. In contrast, feature selection at the classification stage (as claimed) occurs *after* a classifier has already been "trained." (EX-2001 at ¶¶85-86.)

To argue that Tahan discloses the "selectively turned on or off" limitation, Petitioner block quotes the Introduction's description of "Feature Selection" and the criteria used to perform it. (Petition at 20 (block quoting EX-1003 at 3).) According to Petitioner, this quotation "describes ... selecting or reducing the features in the feature vector being input to the ML classifier." (Petition at 21.)

That argument is wrong because, as discussed above, the block quote from

Tahan's Introduction describes "feature selection" that occurs at the *training* stage, not at the classification stage, as claimed.

ii. Tahan's Mal-ID Basic

Tahan's disclosure of Mal-ID Basic also does not describe an ML malware classifier that turns features on or off at the classification stage. Indeed, Mal-ID Basic does not perform feature selection at all.

As shown above (*supra* Section III.B.1.i), Mal-ID Basic proceeds in two phases, each roughly akin to the training stage and the classification stage: (1) a "setup phase," in which data is gathered from a training set of executables, and (2) a "detection phase," in which the data gathered in the setup phase is used in classifying "a previously unseen" executable. (EX-1003 at 7.) During the detection phase, where Mal-ID Basic classifies a "previously unseen executable," the system simply (1) divides the executable into fixed-length segments and then (2) counts how many of those segments meet a given set of criteria. (EX-1003 at 10.)

The detection phase thus does not turn any features "off," within the meaning of the claims. To the extent the "segments" extracted from the executable during that phase are the purported "features," Mal-ID Basic never turns such features "off" because it extracts and analyzes *every* such feature to determine how many meet the set of predefined criteria. Mal-ID Basic does not turn any segment "off," such that it

is not analyzed.

Petitioner argues that each “segment” extracted from the executable is a “feature,” and that Mal-ID Basic turns the feature “off” when it determines that the corresponding segment does not meet the predefined criteria. (*See* Petition at 13 (“Tahan discloses extracting file segments from an executable as ‘features’ ...”); *id.* at 21 (“Tahan teaches selecting segments (features/static data points) ... where segments/features are retained or discarded based on whether an entropy value of the segment is within a certain range (corresponding to ‘one or more features of the feature vector are selectively turned on or off based on whether a value of one or more static data points ... is within a predetermined range.’).)

That argument is wrong because Mal-ID Basic does not turn a segment “off” in response to determining that the segment fails to meet the predefined criteria. As explained above, turning features “off” at the classification stage improves performance of the classification stage by allowing the classification stage to extract and/or analyze fewer than the full set of features identified at the training stage. (EX-2001 at ¶88.) Mal-ID Basic has no such capability. Instead, it always extracts and analyzes every segment (purported “feature”) of the executable. If it determines that a segment meets the predefined criteria, it increments the count of such segments, and if it determines otherwise, it does not increment the count. The performance is the

same either way, so the segments that fail to meet the criteria are not turned “off” within the meaning of the claims.

The same is true for any other potential “feature” Mal-ID Basic extracts from the executable being classified. Contrary to Petitioner’s mapping of Tahan’s “segments” to the recited “features,” Tahan describes the executable’s 3-grams as the extracted “features,” and the “entropy,” “MFG,” and “Spread” measures calculated from those 3-grams as “meta-features.” (EX-1003 at 9.)

Even if those “features” or “meta-features” were mapped to the claimed “features,” Mal-ID Basic would still not teach turning features on or off at the classification stage because, at the detection phase, Mal-ID basic considers *all* of those “features” and “meta-features” to determine how many of the executable’s segments meet the predefined criteria. Because Mal-ID Basic considers all of these features, none is turned “off.” (EX-2001 at ¶88.)

Thus, like Tahan’s Introduction section, Tahan’s disclosure of Mal-ID Basic does not suggest turning any features “on or off” at the classification stage. The combination of those two systems therefore cannot suggest that limitation, and Petitioner does not make any discernable contrary argument.

iii. Tahan’s Mal-ID with ML

When arguing that Tahan turns features on and off, Petitioner does not cite

any portion of Tahan describing Mal-ID with ML. Consequently, the petition fails to show a reasonable likelihood that Mal-ID with ML (whether alone or in combination with Tahan's other disclosures) would have rendered that limitation obvious.

Petitioner could not have argued that Mal-ID with ML makes this disclosure because, like Tahan's Introduction, Mal-ID with ML performs feature selection only at the training stage. As discussed above, the "setup phase" of Mal-ID with ML (akin to a training stage) creates a fixed-length feature vector for each executable in the training set. *See supra* Section III.B.1.iii. That feature vector defines the features that are ultimately extracted from the executable at the detection phase (akin to the classification stage). Tahan does not suggest that Mal-ID with ML may turn any of those features "off" at the classification stage at all, let alone based on the values of static data points extracted from the executable being classified at that stage. (EX-2001 at ¶¶89-90.)

Thus, like Tahan's Introduction and Mal-ID Basic, Mal-ID with ML does not suggest turning any features "on or off" at the classification stage, and the petition contains no contrary argument. *See Koninklijke Philips v. Google*, 948 F.3d 1330, 1336 (Fed. Cir. 2020) (holding that the Board may not institute review based on an argument not raised in the petition).

iv. The combination of Tahan's embodiments

As discussed above, none of the three embodiments disclosed in Tahan suggest turning features on or off at the classification stage—not the traditional ML algorithms of Tahan's Introduction, Mal-ID Basic, or Mal-ID with ML. Consequently, the combination of those embodiments also cannot render that limitation obvious.

Petitioner fails to make any contrary showing. The petition's discussion of the "turned on or off" limitation (Petition at 20-22) cites the "feature selection" algorithm of Tahan's Introduction (*id.* at 20-21 (citing EX-1003 at 3)), but as discussed above, that feature selection occurs at the training stage. Petitioner also cites Tahan's disclosure that, in its "detection phase," Mal-ID Basic counts the number of segments whose entropy is within a given range (*id.* at 21 (citing EX-1003 at 9-10)), but as discussed above, counting the number of segments (purported "features") that meet certain criteria is not turning the feature "on or off"—rather, it is considering and analyzing those features.

Petitioner also fails to acknowledge that the cited features come from *different embodiments*, let alone explain how those embodiments would be combined. This leaves Patent Owner and the Board guessing. Is Petitioner suggesting that the training-stage feature selection disclosed in Tahan's introduction in some way considers

the entropy values used by Mal-ID Basic's detection phase, or is Petitioner suggesting that the detection-phase segment-counting of Mal-ID Basic somehow incorporate some unspecified, training-stage feature-selection ideas from Tahan's introduction? Petitioner does not even acknowledge that it cites different Tahan embodiments, let alone identify what combination Petitioner is proposing.

Nor does Petitioner explain why a POSITA would have been motivated to make the combination or why the POSITA would have had a reasonable expectation of success. Both showings are legally required for a combination argument. *See Eli Lilly & Co. v. Teva Pharms. Int'l GmbH*, 8 F.4th 1331, 1344 (Fed. Cir. 2021) (discussing the "patent challenger's burden to prove that a skilled artisan would have been motivated to combine prior art references" and the separate requirement "that the challenger has also met its burden of showing a reasonable expectation of success").

This problem of intermixing elements of Tahan's different embodiments without acknowledgement or explanation pervades the petition. To illustrate, start with the first limitation ("extracting ... static data points"): Petitioner cites *Mal-ID Basic*'s extraction of fixed-length segments (Petition at 13 (citing EX-1003 at 7-10)), but then attempts to supplement that disclosure with the *Introduction*'s disclosure of extracting other types of features (*id.* at 14 (citing EX-1002 at ¶¶162-167)). For the

next limitation (“generating a feature vector”), Petitioner cites elements of three different embodiments: (1) the *Introduction* (Petition at 16-17 (citing EX-1003 at 2-3; EX-1002 at ¶¶100-102)); (2) *Mal-ID Basic* (Petition at 17 (citing EX-1002 at ¶218 and EX-1003 at 7)); and *Mal-ID with ML* (Petition at 16-17 (citing EX-1003 at 12-13; EX-1002 at ¶¶112-113)).

That the “challenge treats the various features of [Tahan’s] different embodiments as if they are interchangeable with one another” is alone sufficient reason to deny institution. *Abiomed, Inc., et al., v. Maquet Cardiovascular, LLC*, IPR2017-01204, Paper 9 at 9-11 (PTAB Oct. 23, 2017). As in *Abiomed*, the “Petitioner cites a combination of [a prior art reference’s] embodiments” but “fail[s] to establish sufficiently that the features of those different embodiments are interchangeable or provide sufficient rationale to combine the teachings of those different embodiments.” *Id.*

The intermixing of elements in this haphazard and unexplained way renders the petition unclear and frequently nonsensical.

v. The combination of Tahan and Menahem

Menahem does not remedy Tahan’s shortcomings. Petitioner briefly argues that Menahem teaches “selecting features” based on various criteria, such as term frequency (“TF-IDF”). (Petition at 21-22.) But the disclosure Petitioner cites—

Section 3 of Menahem (EX-1004 at 6-7)—discloses only traditional feature selection during “[t]he training process,” not during the classification stage.

As the outline of Menahem's Section 3 explains, Sections 3.2 and 3.3 of Menahem “describe the *training process* and the datasets we had used.” (EX-1004 at 5 (emphasis added).) Section 3.2 discloses “trimming the number of features” *at the training stage*, by “identify[ing] those terms that appear most frequently in files” of the training set. (EX-1004 at 5.) Indeed, the disclosure of Section 3.2 *precedes* the description of “[t]he training process” in Section 3.3, which details “the training of the base-classifiers.” (EX-1004 at 6.) Thus, Section 3 of Menahem does not describe turning features on or off at the classification stage. (EX-2001 at ¶¶92.)

Petitioner also cites to wide swaths of its expert declaration without even a parenthetical to explain why those portions might show that Menahem teaches feature selection at the classification stage. (Petition at 22 (citing EX-1002 at ¶¶130-149, 247-259).) And indeed, none of the cited material does.

The first range of paragraphs cited (*id.* at ¶¶130-149) argues only that “[a] POSITA would have been motivated to combine Tahan's Mal-ID classifier ... with other classifiers into an ensemble as taught by Menahem” (*id.* at ¶141), and to perform “feature selection/reduction” as taught by both references (*id.* at ¶¶145-146). But, as discussed above, the feature selection taught by both references is at the

training stage, not at the classification stage, as claimed.

The second range of paragraphs cited (*id.* at ¶¶247-259) again discusses Tahan's and Menahem's teachings of feature selection at the training stage. Again, those teachings do not suggest selecting features at the classification stage, as claimed.

Petitioner also fails to explain the proposed combination of Tahan and Menahem. As detailed above, Petitioner relies on at least three different embodiments of Tahan, and it is unclear which of these Petitioner proposes should be incorporated into Menahem's ensemble, let alone how or why. Petitioner does not show that *all* classifiers can be plugged into Menahem's ensemble without special considerations, and indeed, it is not clear how either of Tahan's Mal-ID Basic and Mal-ID with ML would use feature vectors created by another classifier, as a combination with Menahem's ensemble would have to. Consequently, Petitioner also fails to show how or why any of Tahan's embodiments could and would be used in Menahem's ensembles. (EX-2001 at ¶¶103-104.)

* * *

As shown above, neither Tahan nor Menahem suggest an ML malware classifier with the flexibility to select features at the classification stage, as required by all the challenged claims. Accordingly, the combination of Tahan and Menahem

does not render any of the challenged claims obvious.

2. The cited combination does not extract static data points that “represent predefined character strings.”

The combination of Tahan and Menahem also fails to render the challenged claims obvious for a second, independent reason: Tahan does not suggest extracting static data points that represent predefined character strings, and Petitioner fails to explain how Tahan could be modified to use that idea or why a POSITA would have been motivated to make that modification.

i. Tahan does not teach extracting “predefined character strings” or “static data points ... [that] represent predefined character strings.”

The independent claims all require “extracting” (from the executable) either “predefined character strings” (independent claim 17) or “static data points ... [that] represent predefined character strings” (independent claims 1 and 10). For example, a static data point may correspond to a field of a PE header, which could include a field name such as “File Size” (*i.e.*, a “predefined character string”) and value, such as 4982784. (EX-1001 at 8:53.) Thus, as the applicant explained during examination, the recited “extracting” steps require analyzing the executable to “identify the predefined character strings and the values associated therewith.” (EX-1007 at 156.)

During prosecution, the applicant introduced this limitation on the extracting

step specifically to overcome a rejection based on U.S. Patent No. 8,709,924 (“Wojnowicz”), which operated like Tahan: it “split [the executable] into non-overlapping file chunks of fixed length” and then performed an “entropy” analysis on those chunks. (*Id.*) The applicant explained that Wojnowicz’s decomposition of files into “fixed-length file chunks” does not satisfy the new limitation respecting a “predefined character strings” because “fixed-length file chunks do not represent predefined character strings.” (*Id.*) Instead, they are “random character string[s],” and extracting them “does not include comparison of the random character string to a predefined character string.” (*Id.*)

The examiner accepted the argument and consequently dropped Wojnowicz as a basis for rejection. (EX-1007 at 18.) Thus, the Patent Office has accepted that “split[ting an executable] into non-overlapping file chunks of fixed length” does not satisfy the claimed limitation of “extracting” either “predefined character strings” (as recited in independent claim 17) or “static data points ...[that] represent predefined character strings” (as recited in independent claims 1 and 10). (EX-1007 at 156; EX-2001 at ¶¶96-98.)

Nevertheless, Petitioner now argues that Tahan satisfies the “extracting” limitations by performing the same operation as Wojnowicz: splitting the executable into fixed-length chunks. (Petition at 13-16.) Specifically, Petitioner cites Tahan’s

description of Mal-ID Basic (*id.* at 13 (citing EX-1003 at 7-10)), wherein the “detection phase” requires “[d]ivid[ing] file F into S segments of length L.” (EX-1003 at 10.)

Board precedent reflects “a commitment to defer to previous Office evaluations of the evidence of record unless material error is shown.” *Advanced Bionics, Oticon Medical AB v. Cochlear Ltd.*, IPR2019-001469, Paper No. 6, at 9 (February 13, 2020) (precedential). This means declining to institute when “substantially the same arguments previously were presented to the Office” and the petitioner has not “demonstrated that the Office erred,” such as by “misapprehending or overlooking specific teachings of the relevant prior art.” *Id.* at 8.

That is the case here. Petitioner presents “substantially the same arguments” respecting the “extracting” step that were previously presented to and rejected by the Office—*i.e.*, that dividing an executable into fixed-length chunks satisfies the “extracting” steps—and Petitioner presents no argument that the Office erred in its previous evaluation of this issue. (Petition at 13-16.) The petition should therefore be denied.

Rather than argue that the Office erred, Petitioner argues that the issue presented here is not the same as that previously considered by the Office because Tahan does not operate like Wojnowicz. (*Id.* at 15.) Specifically, Petitioner reasons that

Tahan decomposes files into relatively *long* chunks (*e.g.*, 64-bytes), which makes it more likely that “the segments extracted by ... Tahan are meaningful portions ... that correspond to computer code, functions, or values.” (*Id.* at 15.)

But the issue is not whether Tahan's segments are more likely than Wojnowicz's to include useful information—it is whether the segments “represent *predefined* character strings,” as claimed. They do not. Tahan's segmentation of the executable is agnostic to the content of those segments. The first step of the “detection phase” is merely to “[d]ivide file F into S segments of length L,” without regard to where those segments start or end. “The Mal-ID implementation uses 2000 segments that are 64-bytes in length.” (EX-1003 at 10.) Tahan does not disclose that the 64-byte segments are chosen specifically to represent some “*predefined* character string,” as claimed, so if one of those segments happens to include some interesting “computer code, functions, or values,” that is mere happenstance.⁴

⁴ In addition to decomposing the executable into fixed-length chunks at the “detection phase” (*i.e.*, classification stage) Tahan also decomposes the files in the training

In short, Tahan does not suggest extracting “predefined character string[s]” or “static data points ... [that] represent predefined character strings,” as claimed.

ii. Petitioner fails to explain how or why a POSITA would modify Tahan to extract “predefined character strings” or “static data points ... [that] represent predefined character strings.”

Menahem does not bridge Tahan’s shortcomings with respect to the “extracting” steps. Petitioner argues that Menahem discloses using an executable’s “portable executable (PE) headers” as features, which Petitioner presumably believes are “predefined character strings” or “static data points ... [that] represent predefined character strings,” as claimed (Petitioner does not explain). (Petition at 14-15.)

The problem is that Petitioner does not explain (1) how Tahan’s Mal-ID system would be modified to use PE headers as the features, (2) why a POSITA would have been motivated to make that modification, or (3) why a POSITA would have

set into fixed-length chunks during the “setup phase” (*i.e.*, training stage) (*See* EX-1003 at 9 (“the file is divided into 64 equal length segments”; “The Mal-ID implementation has used 64-byte segments”).)

had a reasonable expectation of success.

First, Petitioner fails to explain *how* Tahan would be modified to use PE headers (rather than fixed-length segments) as features. Petitioner does not even explain *which* of the systems disclosed in Tahan (that of the Introduction, Mal-ID Basic, Mal-ID with ML, or some other system) Petitioner is proposing should be modified to use PE headers, let alone how. It is not for Patent Owner or the Board to guess at the combination Petitioner proposes: the Board may not institute review based on a combination not raised in the petition. *See Philips*, 948 F.3d at 1336.

Second, Petitioner makes no effort to explain *why* a POSITA would have been motivated to make this unspecified modification to the unspecified system of Tahan. (*See* Petition at 14-15.) It is impossible for Patent Owner to guess at a POSITA's motivation to make the proposed combination because Petitioner has not even articulated the proposed combination. That Petitioner has failed to articulate any motivation to combine is alone sufficient to sink the petition. *See Eli Lilly & Co. v. Teva Pharms. Int'l GmbH*, 8 F.4th 1331, 1344 (Fed. Cir. 2021) (it is the "patent challenger's burden to prove that a skilled artisan would have been motivated to combine prior art references").

Third, Petitioner also fails to explain why a POSITA would have had a reasonable expectation of success in the unspecified combination. "A finding by the

Board that a patent challenger has demonstrated a motivation to combine references does not necessarily imply that the challenger has also met its burden of showing a reasonable expectation of success.” *Id.* And here, Petitioner makes no effort to show that a POSITA would have expected that using PE headers to replace, supplement, or otherwise modify (the Petition does not say which) the “segments” described in Tahan’s Introduction, Mal-ID basic, or Mal-ID with ML (the Petition does not say which) would have successfully improved that system.

For these three reasons, Petitioner fails to show that Menahem’s teaching of using of PE headers as features would have rendered it obvious for Tahan to extract “predefined character strings” or “static data points ... [that] represent predefined character strings” from the executable being classified.

* * *

In sum, the combination of Tahan and Menahem failed to render any of the challenged independent claims obvious because (1) the combination fails to suggest a malware classifier wherein “features of the feature vector are electively turned on or off” *at the classification stage* (based on the file being classified), and (2) Petitioner fails to explain how or why the combination would select features (at any stage) based on “predefined character strings.” Because Petitioner has failed to establish a reasonable likelihood of success on any of the independent claims, it cannot

meet its burden for the challenged dependent claims 2-5, 8, and 11-14 obvious. *Twin-strand Biosciences, Inc. v. Guardant Health, Inc.*, IPR2022-01115, Paper 14, at 19 (PTAB Dec. 30, 2022). Accordingly, Petitioner has failed to show a reasonable likelihood of prevailing on Ground 1, with respect to any of the challenged claims.

D. Ground 2: The combination of Tahan, Menahem, and Schultz did not render dependent claims 6, 7, 9, 15, 16, 18, and 20 obvious.

Ground 2 fails for similar reasons as Ground 1 because it adds Schultz only to purportedly teach the limitations *added* by dependent claims 6, 7, 9, 15, 16, 18, and 20. (Petition at 55-69.) Because Ground 2 includes no argument that Schultz remedies any of Ground 1's shortcomings respecting the independent claims, and the claims it challenges all depend on those independent claims, Ground 2 cannot render any of the challenged claims obvious.

IV. THERE ARE OBJECTIVE INDICIA OF NON-OBVIOUSNESS

In determining obviousness of the claimed invention, objective evidence of secondary considerations pertaining to nonobviousness must be considered, including long-felt but unsolved needs, failure of others, and commercial success. *In re Huang*, 100 F.3d 135, 139 (Fed. Cir. 1996) (citing *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966)). These indicia constitute independent evidence of nonobviousness that can be the most probative evidence in that regard “in avoiding the trap of

hindsight when reviewing, what otherwise seems like, a combination of known elements.” *Leo Pharm. Prods., Ltd. v. Rea*, 726 F.3d 1346, 1358 (Fed. Cir. 2013).

The Board has repeatedly cautioned “that known evidence of secondary considerations should be addressed in the petition.” *See, e.g., Robert Bosch Tool Corp. v. SD3, LLC*, IPR2016-01751, Paper 15, at 23 (PTAB Mar. 22, 2017) (collecting prior Board decisions that deny institution based, at least in part, on a petitioner’s failure to address known evidence of secondary considerations). Despite Patent Owner providing evidence regarding the long-felt, unmet need for the invention in its district court complaint against Petitioner (EX-2011 at 23-26), Petitioner fails to consider such evidence in its petition; thus, institution should be denied.

As discussed above, a long-felt and persistent need existed to improve malware classification performance by dynamically calibrating feature vectors at the classification stage based on the content of the executable. (EX-2001 at ¶109.) Prior art attempts were unable to improve classification performance by performing such dynamic calibration at the classification stage. Where others failed, the inventors of the ’844 patent succeeded.

Patent Owner currently accuses at least four industry participants, including Petitioner, of infringing the innovative technology claimed in the ’844 patent, demonstrating the critical need and widespread industry reliance on these novel

methods. (EX-2011 at 113-118.) Petitioner, for example, markets and provide Apex One and its EDR/XDR products, the components of which practice the systems and methods claimed in the '844 patent.

Accordingly, secondary objective indicia of non-obviousness confirm that Petitioner cannot show a reasonable likelihood of success as to any of the challenged independent claims.

V. CONCLUSION

For the foregoing reasons, the Board should decline to institute an IPR of the '844 patent.

Date: July 17, 2023

Respectfully submitted,

BY: /s/ Brian Eutermoser
Brian Eutermoser
Registration No. 64,058

CERTIFICATION OF WORD COUNT

The undersigned hereby certifies that the portions of the above-captioned **PA-
TENT OWNER'S PRELIMINARY RESPONSE** specified in 37 C.F.R. § 42.24 includes 12,941 words and therefore complies with the 14,000-word limit set forth in 37 C.F.R. § 42.24. This word count was prepared using Microsoft Word 365 v.2202.

Date: July 17, 2023

Respectfully submitted,

BY: /s/ *Brian Eutermoser*
Brian Eutermoser
Registration No. 64,058

Certificate of Service

The undersigned hereby certifies that the foregoing **PATENT OWNER'S PRELIMINARY RESPONSE** and all supporting **EXHIBITS 2001-2017** were served by filing the documents through the Patent Trial and Appeal Case Tracking System (P-TACTS) and courtesy copies were also served via electronic mail, in their entireties on the following:

Robert M. Hansen
Matthew J. Anderson
Scott B. Amankwatia
The Marbury Law Group, PLLC
11800 Sunrise Valley Drive, 15th Floor
Reston, VA 20191
rhansen@marburylaw.com
manderson@marburylaw.com
samankwatia@marburylaw.com

Date: July 17, 2023

KING & SPALDING LLP

BY: /s/ Brian Eutermoser
Brian Eutermoser
Registration No. 64,058

Attorney for Patent Owner *Open Text Inc.*