



Federated Learning with CLARA

李柏穎

110年7月6日

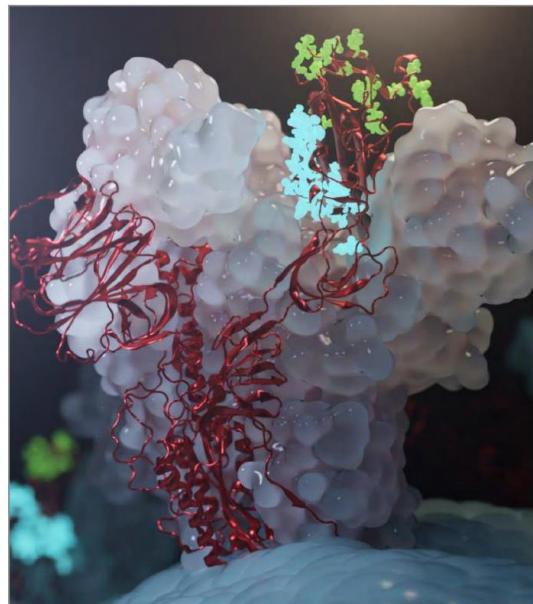
Agenda

- Clara 介紹
- Clara Train SDK 4.0
- Clara 在TWCC平台配置流程
- Clara 測試
- Clara在NCHC 的應用

Clara 介紹

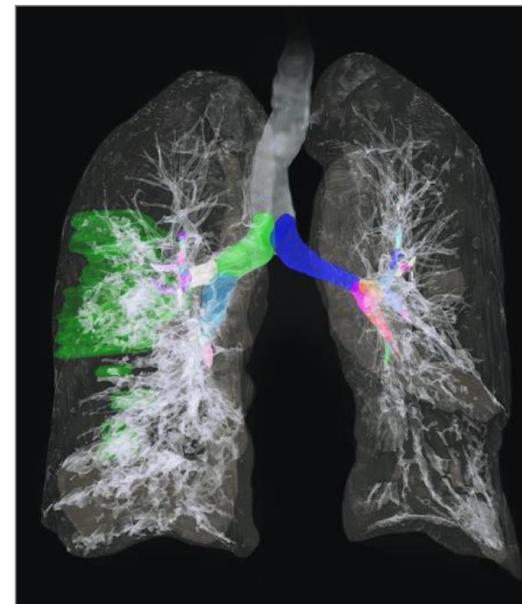
NVIDIA CLARA APPLICATION FRAMEWORKS

Drug Discovery



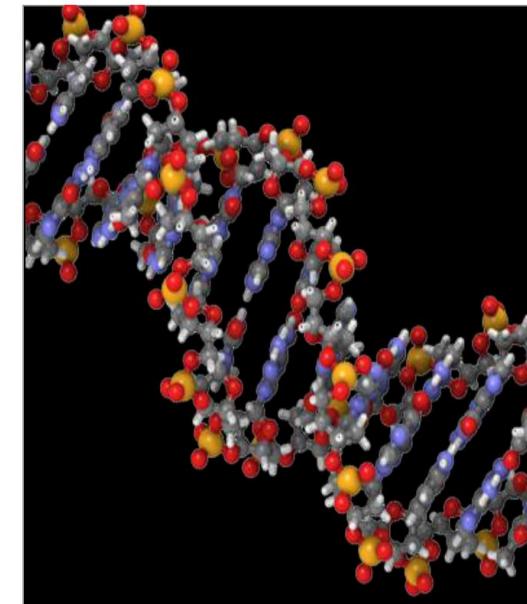
[NVIDIA Clara Discovery](#)

Medical Imaging



[NVIDIA Clara Imaging](#)

Accelerated Genomics



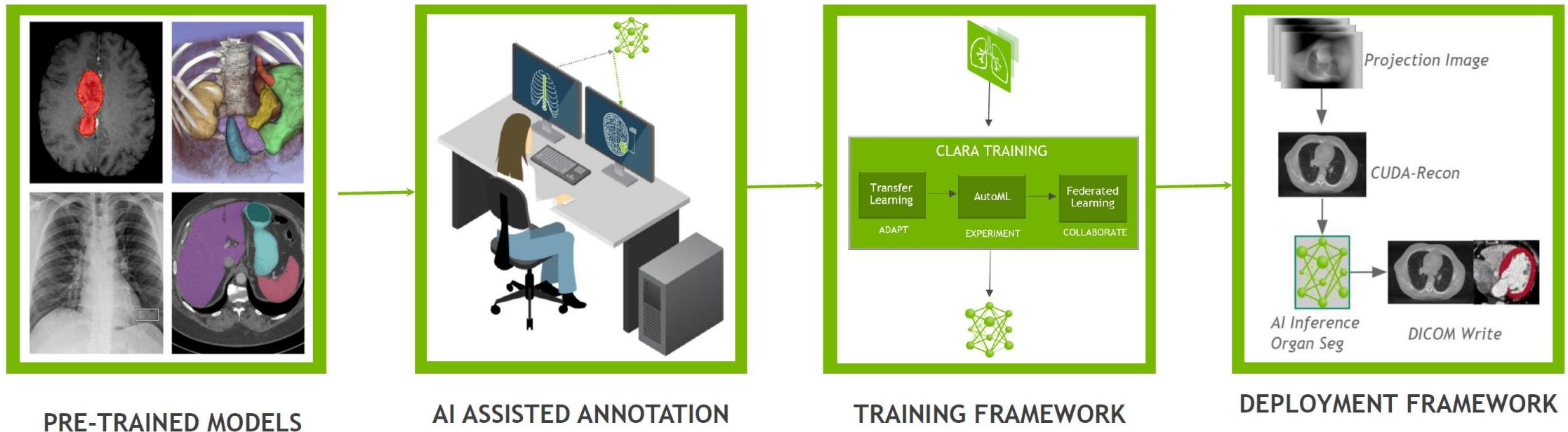
[NVIDIA Clara Parabricks](#)

Smart Hospitals

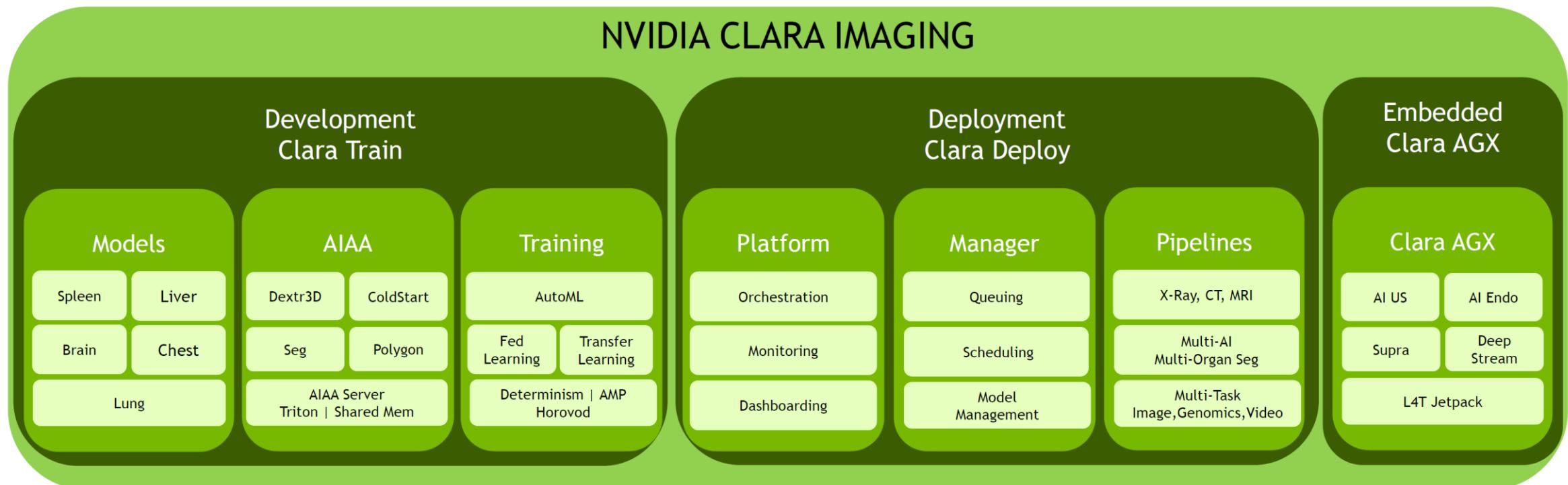


[NVIDIA Clara Guardian](#)

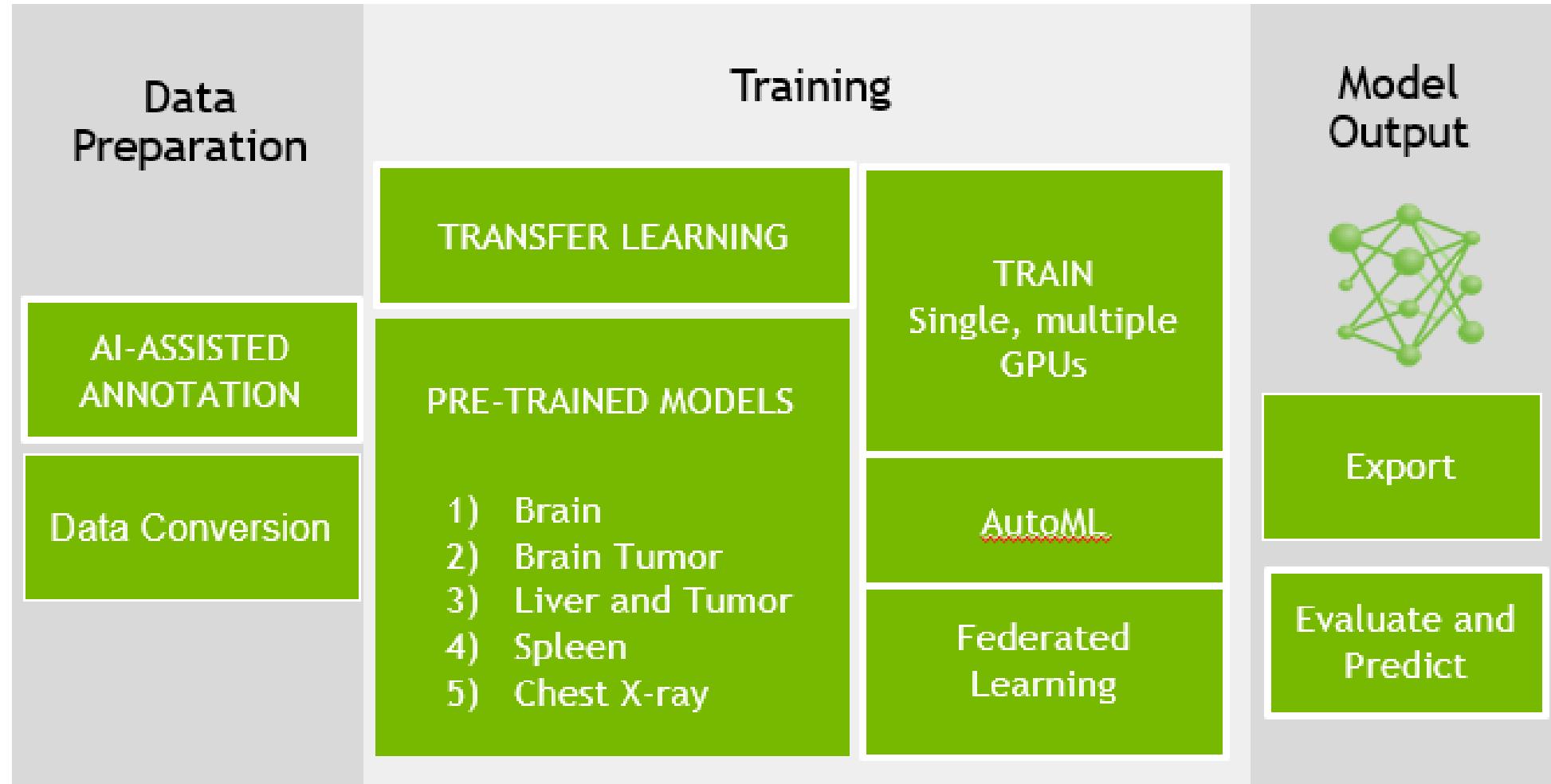
Nvidia Clara Imaging



Nvidia Clara Imaging



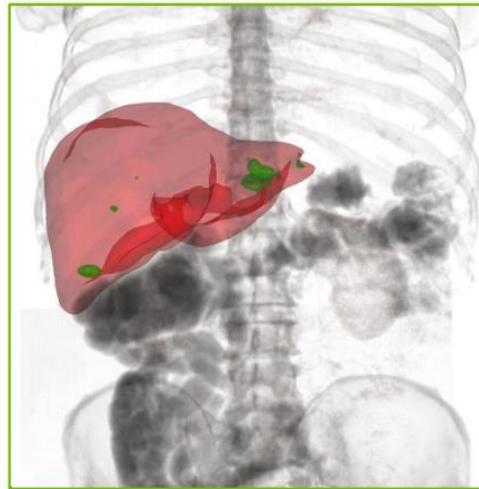
Clara Train SDK



Clara Train SDK

- Clara Training Framework
- AI-Assisted Annotation (AIAA)
- AutoML
- Federated learning

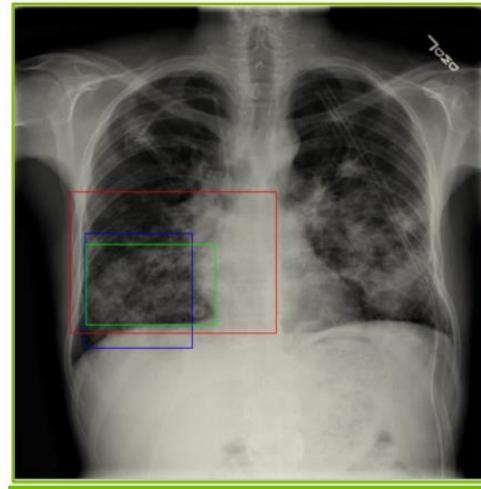
Clara Training Framework – Pre-trained Models



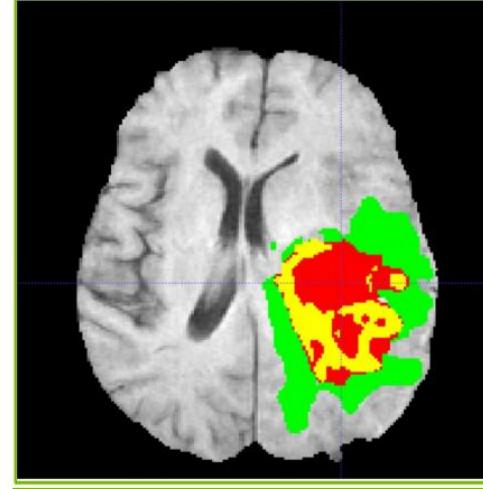
Liver Tumor Segmentation



Spleen Segmentation



Chest X-Ray Classification



Brain Tumor Segmentation

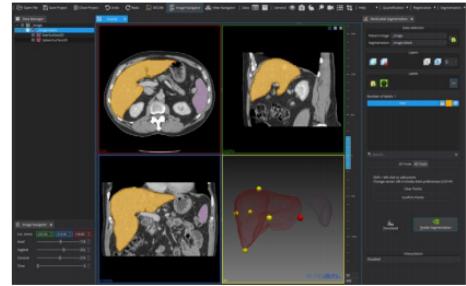
Model	Medical Task	Data	Network
Brain tumor segmentation	3D Segmentation	MR (BraTS 2018)	Res-UNet
Liver and tumor segmentation	3D Segmentation	CT (medical Decath)	Anisotropic Hybrid Network (AH-Net)
Spleen	3D Segmentation	CT (MSD)	Network (AH-Net)
Chest X-ray classification	2D Classification	PLCO	DenseNet121
COVID-19 Segmentation & Classification	3D Segmentation & Classification	CT	AH-Net + DenseNet

Clara Training Framework – Medical Model ARchive (MMAR)

NARLabs

```
7      "train":  
8      {  
9          "loss":  
10         {"name": "Dice"},  
11  
12  
13  
14         "optimizer":  
15         {"name": "Adam"},  
16  
17  
18         "lr_policy":  
19         {"name": "DecayLRonStep"},  
20  
21  
22  
23  
24  
25  
26  
27  
28         "model":  
29         {  
30             "name": "SegmAhnet3D", ←  
31             "args": {  
32                 "num_classes": 3,  
33                 "if_from_scratch": false,  
34                 "if_use_psp": false,  
35                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
36                 "plane": "z",  
37                 "final_activation": "softmax"  
38             }  
39         },  
40  
41         "pre_transforms":  
42         [...],  
43  
44         "image_pipeline": {"name": "ImagePipeline"}  
45     },  
46  
47     "val":  
48     {  
49         "loss":  
50         {"name": "Dice"},  
51  
52         "optimizer":  
53         {"name": "Adam"},  
54  
55         "lr_policy":  
56         {"name": "DecayLRonStep"},  
57  
58         "model":  
59         {  
60             "name": "SegmAhnet3D", ←  
61             "args": {  
62                 "num_classes": 3,  
63                 "if_from_scratch": false,  
64                 "if_use_psp": false,  
65                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
66                 "plane": "z",  
67                 "final_activation": "softmax"  
68             }  
69         },  
70  
71         "pre_transforms":  
72         [...],  
73  
74         "image_pipeline": {"name": "ImagePipeline"}  
75     },  
76  
77     "test":  
78     {  
79         "loss":  
80         {"name": "Dice"},  
81  
82         "optimizer":  
83         {"name": "Adam"},  
84  
85         "lr_policy":  
86         {"name": "DecayLRonStep"},  
87  
88         "model":  
89         {  
90             "name": "SegmAhnet3D", ←  
91             "args": {  
92                 "num_classes": 3,  
93                 "if_from_scratch": false,  
94                 "if_use_psp": false,  
95                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
96                 "plane": "z",  
97                 "final_activation": "softmax"  
98             }  
99         },  
100  
101         "pre_transforms":  
102         [...],  
103  
104         "image_pipeline": {"name": "ImagePipeline"}  
105     },  
106  
107     "postprocess":  
108     {  
109         "loss":  
110         {"name": "Dice"},  
111  
112         "optimizer":  
113         {"name": "Adam"},  
114  
115         "lr_policy":  
116         {"name": "DecayLRonStep"},  
117  
118         "model":  
119         {  
120             "name": "SegmAhnet3D", ←  
121             "args": {  
122                 "num_classes": 3,  
123                 "if_from_scratch": false,  
124                 "if_use_psp": false,  
125                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
126                 "plane": "z",  
127                 "final_activation": "softmax"  
128             }  
129         },  
130  
131         "pre_transforms":  
132         [...],  
133  
134         "image_pipeline": {"name": "ImagePipeline"}  
135     },  
136  
137     "postprocess":  
138     {  
139         "loss":  
140         {"name": "Dice"},  
141  
142         "optimizer":  
143         {"name": "Adam"},  
144  
145         "lr_policy":  
146         {"name": "DecayLRonStep"},  
147  
148         "model":  
149         {  
150             "name": "SegmAhnet3D", ←  
151             "args": {  
152                 "num_classes": 3,  
153                 "if_from_scratch": false,  
154                 "if_use_psp": false,  
155                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
156                 "plane": "z",  
157                 "final_activation": "softmax"  
158             }  
159         },  
160  
161         "pre_transforms":  
162         [...],  
163  
164         "image_pipeline": {"name": "ImagePipeline"}  
165     },  
166  
167     "postprocess":  
168     {  
169         "loss":  
170         {"name": "Dice"},  
171  
172         "optimizer":  
173         {"name": "Adam"},  
174  
175         "lr_policy":  
176         {"name": "DecayLRonStep"},  
177  
178         "model":  
179         {  
180             "name": "SegmAhnet3D", ←  
181             "args": {  
182                 "num_classes": 3,  
183                 "if_from_scratch": false,  
184                 "if_use_psp": false,  
185                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
186                 "plane": "z",  
187                 "final_activation": "softmax"  
188             }  
189         },  
190  
191         "pre_transforms":  
192         [...],  
193  
194         "image_pipeline": {"name": "ImagePipeline"}  
195     },  
196  
197     "postprocess":  
198     {  
199         "loss":  
200         {"name": "Dice"},  
201  
202         "optimizer":  
203         {"name": "Adam"},  
204  
205         "lr_policy":  
206         {"name": "DecayLRonStep"},  
207  
208         "model":  
209         {  
210             "name": "SegmAhnet3D", ←  
211             "args": {  
212                 "num_classes": 3,  
213                 "if_from_scratch": false,  
214                 "if_use_psp": false,  
215                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
216                 "plane": "z",  
217                 "final_activation": "softmax"  
218             }  
219         },  
220  
221         "pre_transforms":  
222         [...],  
223  
224         "image_pipeline": {"name": "ImagePipeline"}  
225     },  
226  
227     "postprocess":  
228     {  
229         "loss":  
230         {"name": "Dice"},  
231  
232         "optimizer":  
233         {"name": "Adam"},  
234  
235         "lr_policy":  
236         {"name": "DecayLRonStep"},  
237  
238         "model":  
239         {  
240             "name": "SegmAhnet3D", ←  
241             "args": {  
242                 "num_classes": 3,  
243                 "if_from_scratch": false,  
244                 "if_use_psp": false,  
245                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
246                 "plane": "z",  
247                 "final_activation": "softmax"  
248             }  
249         },  
250  
251         "pre_transforms":  
252         [...],  
253  
254         "image_pipeline": {"name": "ImagePipeline"}  
255     },  
256  
257     "postprocess":  
258     {  
259         "loss":  
260         {"name": "Dice"},  
261  
262         "optimizer":  
263         {"name": "Adam"},  
264  
265         "lr_policy":  
266         {"name": "DecayLRonStep"},  
267  
268         "model":  
269         {  
270             "name": "SegmAhnet3D", ←  
271             "args": {  
272                 "num_classes": 3,  
273                 "if_from_scratch": false,  
274                 "if_use_psp": false,  
275                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
276                 "plane": "z",  
277                 "final_activation": "softmax"  
278             }  
279         },  
280  
281         "pre_transforms":  
282         [...],  
283  
284         "image_pipeline": {"name": "ImagePipeline"}  
285     },  
286  
287     "postprocess":  
288     {  
289         "loss":  
290         {"name": "Dice"},  
291  
292         "optimizer":  
293         {"name": "Adam"},  
294  
295         "lr_policy":  
296         {"name": "DecayLRonStep"},  
297  
298         "model":  
299         {  
300             "name": "SegmAhnet3D", ←  
301             "args": {  
302                 "num_classes": 3,  
303                 "if_from_scratch": false,  
304                 "if_use_psp": false,  
305                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
306                 "plane": "z",  
307                 "final_activation": "softmax"  
308             }  
309         },  
310  
311         "pre_transforms":  
312         [...],  
313  
314         "image_pipeline": {"name": "ImagePipeline"}  
315     },  
316  
317     "postprocess":  
318     {  
319         "loss":  
320         {"name": "Dice"},  
321  
322         "optimizer":  
323         {"name": "Adam"},  
324  
325         "lr_policy":  
326         {"name": "DecayLRonStep"},  
327  
328         "model":  
329         {  
330             "name": "SegmAhnet3D", ←  
331             "args": {  
332                 "num_classes": 3,  
333                 "if_from_scratch": false,  
334                 "if_use_psp": false,  
335                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
336                 "plane": "z",  
337                 "final_activation": "softmax"  
338             }  
339         },  
340  
341         "pre_transforms":  
342         [...],  
343  
344         "image_pipeline": {"name": "ImagePipeline"}  
345     },  
346  
347     "postprocess":  
348     {  
349         "loss":  
350         {"name": "Dice"},  
351  
352         "optimizer":  
353         {"name": "Adam"},  
354  
355         "lr_policy":  
356         {"name": "DecayLRonStep"},  
357  
358         "model":  
359         {  
360             "name": "SegmAhnet3D", ←  
361             "args": {  
362                 "num_classes": 3,  
363                 "if_from_scratch": false,  
364                 "if_use_psp": false,  
365                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
366                 "plane": "z",  
367                 "final_activation": "softmax"  
368             }  
369         },  
370  
371         "pre_transforms":  
372         [...],  
373  
374         "image_pipeline": {"name": "ImagePipeline"}  
375     },  
376  
377     "postprocess":  
378     {  
379         "loss":  
380         {"name": "Dice"},  
381  
382         "optimizer":  
383         {"name": "Adam"},  
384  
385         "lr_policy":  
386         {"name": "DecayLRonStep"},  
387  
388         "model":  
389         {  
390             "name": "SegmAhnet3D", ←  
391             "args": {  
392                 "num_classes": 3,  
393                 "if_from_scratch": false,  
394                 "if_use_psp": false,  
395                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
396                 "plane": "z",  
397                 "final_activation": "softmax"  
398             }  
399         },  
400  
401         "pre_transforms":  
402         [...],  
403  
404         "image_pipeline": {"name": "ImagePipeline"}  
405     },  
406  
407     "postprocess":  
408     {  
409         "loss":  
410         {"name": "Dice"},  
411  
412         "optimizer":  
413         {"name": "Adam"},  
414  
415         "lr_policy":  
416         {"name": "DecayLRonStep"},  
417  
418         "model":  
419         {  
420             "name": "SegmAhnet3D", ←  
421             "args": {  
422                 "num_classes": 3,  
423                 "if_from_scratch": false,  
424                 "if_use_psp": false,  
425                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
426                 "plane": "z",  
427                 "final_activation": "softmax"  
428             }  
429         },  
430  
431         "pre_transforms":  
432         [...],  
433  
434         "image_pipeline": {"name": "ImagePipeline"}  
435     },  
436  
437     "postprocess":  
438     {  
439         "loss":  
440         {"name": "Dice"},  
441  
442         "optimizer":  
443         {"name": "Adam"},  
444  
445         "lr_policy":  
446         {"name": "DecayLRonStep"},  
447  
448         "model":  
449         {  
450             "name": "SegmAhnet3D", ←  
451             "args": {  
452                 "num_classes": 3,  
453                 "if_from_scratch": false,  
454                 "if_use_psp": false,  
455                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
456                 "plane": "z",  
457                 "final_activation": "softmax"  
458             }  
459         },  
460  
461         "pre_transforms":  
462         [...],  
463  
464         "image_pipeline": {"name": "ImagePipeline"}  
465     },  
466  
467     "postprocess":  
468     {  
469         "loss":  
470         {"name": "Dice"},  
471  
472         "optimizer":  
473         {"name": "Adam"},  
474  
475         "lr_policy":  
476         {"name": "DecayLRonStep"},  
477  
478         "model":  
479         {  
480             "name": "SegmAhnet3D", ←  
481             "args": {  
482                 "num_classes": 3,  
483                 "if_from_scratch": false,  
484                 "if_use_psp": false,  
485                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
486                 "plane": "z",  
487                 "final_activation": "softmax"  
488             }  
489         },  
490  
491         "pre_transforms":  
492         [...],  
493  
494         "image_pipeline": {"name": "ImagePipeline"}  
495     },  
496  
497     "postprocess":  
498     {  
499         "loss":  
500         {"name": "Dice"},  
501  
502         "optimizer":  
503         {"name": "Adam"},  
504  
505         "lr_policy":  
506         {"name": "DecayLRonStep"},  
507  
508         "model":  
509         {  
510             "name": "SegmAhnet3D", ←  
511             "args": {  
512                 "num_classes": 3,  
513                 "if_from_scratch": false,  
514                 "if_use_psp": false,  
515                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
516                 "plane": "z",  
517                 "final_activation": "softmax"  
518             }  
519         },  
520  
521         "pre_transforms":  
522         [...],  
523  
524         "image_pipeline": {"name": "ImagePipeline"}  
525     },  
526  
527     "postprocess":  
528     {  
529         "loss":  
530         {"name": "Dice"},  
531  
532         "optimizer":  
533         {"name": "Adam"},  
534  
535         "lr_policy":  
536         {"name": "DecayLRonStep"},  
537  
538         "model":  
539         {  
540             "name": "SegmAhnet3D", ←  
541             "args": {  
542                 "num_classes": 3,  
543                 "if_from_scratch": false,  
544                 "if_use_psp": false,  
545                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
546                 "plane": "z",  
547                 "final_activation": "softmax"  
548             }  
549         },  
550  
551         "pre_transforms":  
552         [...],  
553  
554         "image_pipeline": {"name": "ImagePipeline"}  
555     },  
556  
557     "postprocess":  
558     {  
559         "loss":  
560         {"name": "Dice"},  
561  
562         "optimizer":  
563         {"name": "Adam"},  
564  
565         "lr_policy":  
566         {"name": "DecayLRonStep"},  
567  
568         "model":  
569         {  
570             "name": "SegmAhnet3D", ←  
571             "args": {  
572                 "num_classes": 3,  
573                 "if_from_scratch": false,  
574                 "if_use_psp": false,  
575                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
576                 "plane": "z",  
577                 "final_activation": "softmax"  
578             }  
579         },  
580  
581         "pre_transforms":  
582         [...],  
583  
584         "image_pipeline": {"name": "ImagePipeline"}  
585     },  
586  
587     "postprocess":  
588     {  
589         "loss":  
590         {"name": "Dice"},  
591  
592         "optimizer":  
593         {"name": "Adam"},  
594  
595         "lr_policy":  
596         {"name": "DecayLRonStep"},  
597  
598         "model":  
599         {  
600             "name": "SegmAhnet3D", ←  
601             "args": {  
602                 "num_classes": 3,  
603                 "if_from_scratch": false,  
604                 "if_use_psp": false,  
605                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
606                 "plane": "z",  
607                 "final_activation": "softmax"  
608             }  
609         },  
610  
611         "pre_transforms":  
612         [...],  
613  
614         "image_pipeline": {"name": "ImagePipeline"}  
615     },  
616  
617     "postprocess":  
618     {  
619         "loss":  
620         {"name": "Dice"},  
621  
622         "optimizer":  
623         {"name": "Adam"},  
624  
625         "lr_policy":  
626         {"name": "DecayLRonStep"},  
627  
628         "model":  
629         {  
630             "name": "SegmAhnet3D", ←  
631             "args": {  
632                 "num_classes": 3,  
633                 "if_from_scratch": false,  
634                 "if_use_psp": false,  
635                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
636                 "plane": "z",  
637                 "final_activation": "softmax"  
638             }  
639         },  
640  
641         "pre_transforms":  
642         [...],  
643  
644         "image_pipeline": {"name": "ImagePipeline"}  
645     },  
646  
647     "postprocess":  
648     {  
649         "loss":  
650         {"name": "Dice"},  
651  
652         "optimizer":  
653         {"name": "Adam"},  
654  
655         "lr_policy":  
656         {"name": "DecayLRonStep"},  
657  
658         "model":  
659         {  
660             "name": "SegmAhnet3D", ←  
661             "args": {  
662                 "num_classes": 3,  
663                 "if_from_scratch": false,  
664                 "if_use_psp": false,  
665                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
666                 "plane": "z",  
667                 "final_activation": "softmax"  
668             }  
669         },  
670  
671         "pre_transforms":  
672         [...],  
673  
674         "image_pipeline": {"name": "ImagePipeline"}  
675     },  
676  
677     "postprocess":  
678     {  
679         "loss":  
680         {"name": "Dice"},  
681  
682         "optimizer":  
683         {"name": "Adam"},  
684  
685         "lr_policy":  
686         {"name": "DecayLRonStep"},  
687  
688         "model":  
689         {  
690             "name": "SegmAhnet3D", ←  
691             "args": {  
692                 "num_classes": 3,  
693                 "if_from_scratch": false,  
694                 "if_use_psp": false,  
695                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
696                 "plane": "z",  
697                 "final_activation": "softmax"  
698             }  
699         },  
700  
701         "pre_transforms":  
702         [...],  
703  
704         "image_pipeline": {"name": "ImagePipeline"}  
705     },  
706  
707     "postprocess":  
708     {  
709         "loss":  
710         {"name": "Dice"},  
711  
712         "optimizer":  
713         {"name": "Adam"},  
714  
715         "lr_policy":  
716         {"name": "DecayLRonStep"},  
717  
718         "model":  
719         {  
720             "name": "SegmAhnet3D", ←  
721             "args": {  
722                 "num_classes": 3,  
723                 "if_from_scratch": false,  
724                 "if_use_psp": false,  
725                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
726                 "plane": "z",  
727                 "final_activation": "softmax"  
728             }  
729         },  
730  
731         "pre_transforms":  
732         [...],  
733  
734         "image_pipeline": {"name": "ImagePipeline"}  
735     },  
736  
737     "postprocess":  
738     {  
739         "loss":  
740         {"name": "Dice"},  
741  
742         "optimizer":  
743         {"name": "Adam"},  
744  
745         "lr_policy":  
746         {"name": "DecayLRonStep"},  
747  
748         "model":  
749         {  
750             "name": "SegmAhnet3D", ←  
751             "args": {  
752                 "num_classes": 3,  
753                 "if_from_scratch": false,  
754                 "if_use_psp": false,  
755                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
756                 "plane": "z",  
757                 "final_activation": "softmax"  
758             }  
759         },  
760  
761         "pre_transforms":  
762         [...],  
763  
764         "image_pipeline": {"name": "ImagePipeline"}  
765     },  
766  
767     "postprocess":  
768     {  
769         "loss":  
770         {"name": "Dice"},  
771  
772         "optimizer":  
773         {"name": "Adam"},  
774  
775         "lr_policy":  
776         {"name": "DecayLRonStep"},  
777  
778         "model":  
779         {  
780             "name": "SegmAhnet3D", ←  
781             "args": {  
782                 "num_classes": 3,  
783                 "if_from_scratch": false,  
784                 "if_use_psp": false,  
785                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
786                 "plane": "z",  
787                 "final_activation": "softmax"  
788             }  
789         },  
790  
791         "pre_transforms":  
792         [...],  
793  
794         "image_pipeline": {"name": "ImagePipeline"}  
795     },  
796  
797     "postprocess":  
798     {  
799         "loss":  
800         {"name": "Dice"},  
801  
802         "optimizer":  
803         {"name": "Adam"},  
804  
805         "lr_policy":  
806         {"name": "DecayLRonStep"},  
807  
808         "model":  
809         {  
810             "name": "SegmAhnet3D", ←  
811             "args": {  
812                 "num_classes": 3,  
813                 "if_from_scratch": false,  
814                 "if_use_psp": false,  
815                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
816                 "plane": "z",  
817                 "final_activation": "softmax"  
818             }  
819         },  
820  
821         "pre_transforms":  
822         [...],  
823  
824         "image_pipeline": {"name": "ImagePipeline"}  
825     },  
826  
827     "postprocess":  
828     {  
829         "loss":  
830         {"name": "Dice"},  
831  
832         "optimizer":  
833         {"name": "Adam"},  
834  
835         "lr_policy":  
836         {"name": "DecayLRonStep"},  
837  
838         "model":  
839         {  
840             "name": "SegmAhnet3D", ←  
841             "args": {  
842                 "num_classes": 3,  
843                 "if_from_scratch": false,  
844                 "if_use_psp": false,  
845                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
846                 "plane": "z",  
847                 "final_activation": "softmax"  
848             }  
849         },  
850  
851         "pre_transforms":  
852         [...],  
853  
854         "image_pipeline": {"name": "ImagePipeline"}  
855     },  
856  
857     "postprocess":  
858     {  
859         "loss":  
860         {"name": "Dice"},  
861  
862         "optimizer":  
863         {"name": "Adam"},  
864  
865         "lr_policy":  
866         {"name": "DecayLRonStep"},  
867  
868         "model":  
869         {  
870             "name": "SegmAhnet3D", ←  
871             "args": {  
872                 "num_classes": 3,  
873                 "if_from_scratch": false,  
874                 "if_use_psp": false,  
875                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
876                 "plane": "z",  
877                 "final_activation": "softmax"  
878             }  
879         },  
880  
881         "pre_transforms":  
882         [...],  
883  
884         "image_pipeline": {"name": "ImagePipeline"}  
885     },  
886  
887     "postprocess":  
888     {  
889         "loss":  
890         {"name": "Dice"},  
891  
892         "optimizer":  
893         {"name": "Adam"},  
894  
895         "lr_policy":  
896         {"name": "DecayLRonStep"},  
897  
898         "model":  
899         {  
900             "name": "SegmAhnet3D", ←  
901             "args": {  
902                 "num_classes": 3,  
903                 "if_from_scratch": false,  
904                 "if_use_psp": false,  
905                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
906                 "plane": "z",  
907                 "final_activation": "softmax"  
908             }  
909         },  
910  
911         "pre_transforms":  
912         [...],  
913  
914         "image_pipeline": {"name": "ImagePipeline"}  
915     },  
916  
917     "postprocess":  
918     {  
919         "loss":  
920         {"name": "Dice"},  
921  
922         "optimizer":  
923         {"name": "Adam"},  
924  
925         "lr_policy":  
926         {"name": "DecayLRonStep"},  
927  
928         "model":  
929         {  
930             "name": "SegmAhnet3D", ←  
931             "args": {  
932                 "num_classes": 3,  
933                 "if_from_scratch": false,  
934                 "if_use_psp": false,  
935                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
936                 "plane": "z",  
937                 "final_activation": "softmax"  
938             }  
939         },  
940  
941         "pre_transforms":  
942         [...],  
943  
944         "image_pipeline": {"name": "ImagePipeline"}  
945     },  
946  
947     "postprocess":  
948     {  
949         "loss":  
950         {"name": "Dice"},  
951  
952         "optimizer":  
953         {"name": "Adam"},  
954  
955         "lr_policy":  
956         {"name": "DecayLRonStep"},  
957  
958         "model":  
959         {  
960             "name": "SegmAhnet3D", ←  
961             "args": {  
962                 "num_classes": 3,  
963                 "if_from_scratch": false,  
964                 "if_use_psp": false,  
965                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
966                 "plane": "z",  
967                 "final_activation": "softmax"  
968             }  
969         },  
970  
971         "pre_transforms":  
972         [...],  
973  
974         "image_pipeline": {"name": "ImagePipeline"}  
975     },  
976  
977     "postprocess":  
978     {  
979         "loss":  
980         {"name": "Dice"},  
981  
982         "optimizer":  
983         {"name": "Adam"},  
984  
985         "lr_policy":  
986         {"name": "DecayLRonStep"},  
987  
988         "model":  
989         {  
990             "name": "SegmAhnet3D", ←  
991             "args": {  
992                 "num_classes": 3,  
993                 "if_from_scratch": false,  
994                 "if_use_psp": false,  
995                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
996                 "plane": "z",  
997                 "final_activation": "softmax"  
998             }  
999         },  
1000  
1001         "pre_transforms":  
1002         [...],  
1003  
1004         "image_pipeline": {"name": "ImagePipeline"}  
1005     },  
1006  
1007     "postprocess":  
1008     {  
1009         "loss":  
1010         {"name": "Dice"},  
1011  
1012         "optimizer":  
1013         {"name": "Adam"},  
1014  
1015         "lr_policy":  
1016         {"name": "DecayLRonStep"},  
1017  
1018         "model":  
1019         {  
1020             "name": "SegmAhnet3D", ←  
1021             "args": {  
1022                 "num_classes": 3,  
1023                 "if_from_scratch": false,  
1024                 "if_use_psp": false,  
1025                 "pretrain_weight_name": "{PRETRAIN_WEIGHTS_FILE}",  
1026                 "plane": "z",  
1027                 "final_activation": "softmax"  
1028             }  
1029         },  
1030  
1031         "pre_transforms":  
1032         [...],  
1033  
1034         "image_pipeline": {"name": "ImagePipeline"}  
1035     },  
1036  
1037     "postprocess":  
1038     {  
1039         "loss":  
1040         {"name": "Dice"},  
1041  
1042         "optimizer":  
1043         {"name": "Adam"},  
1044  
1045        
```

AI-Assisted Annotation (AIAA)



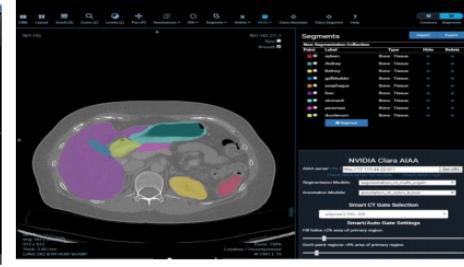
MITK
AIAA client



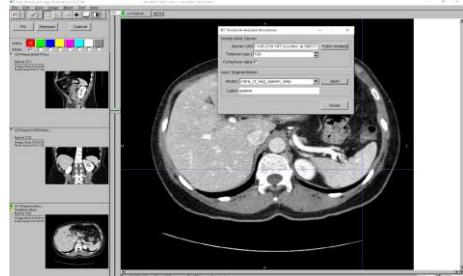
3D slicer
AIAA client



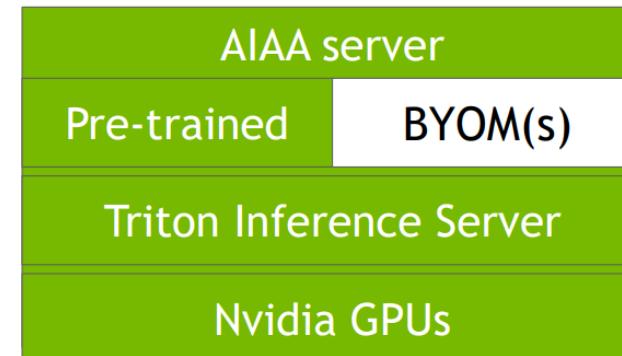
FOVIA
AIAA client



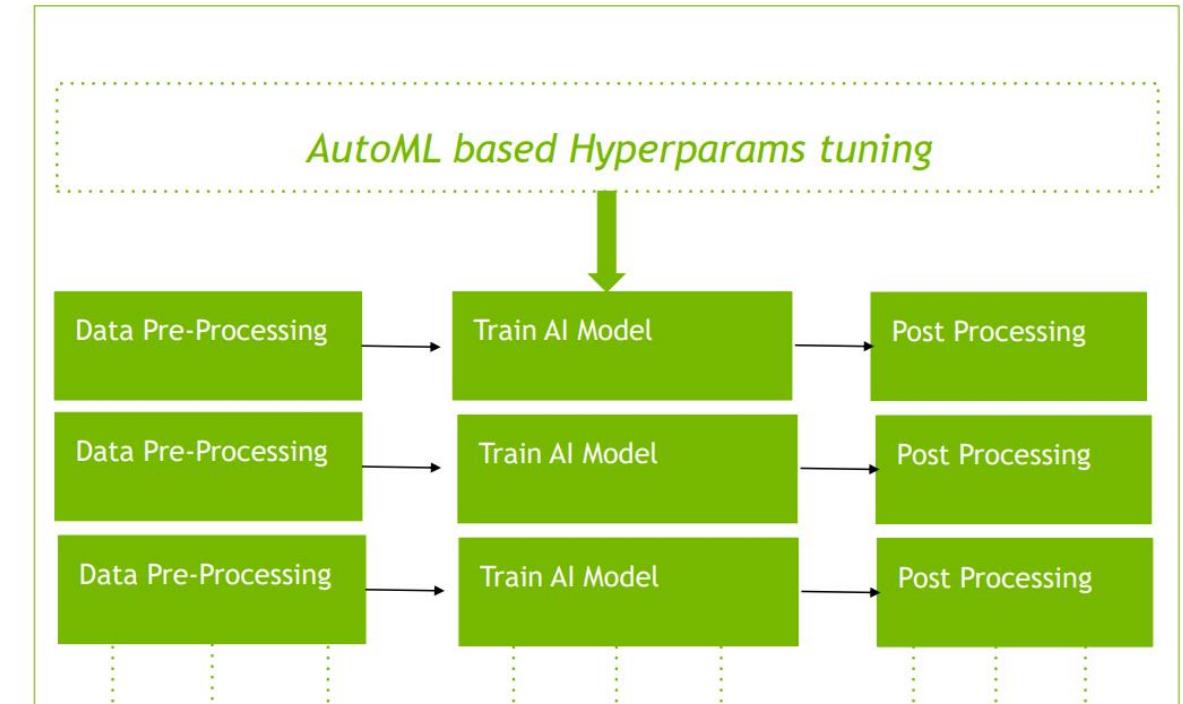
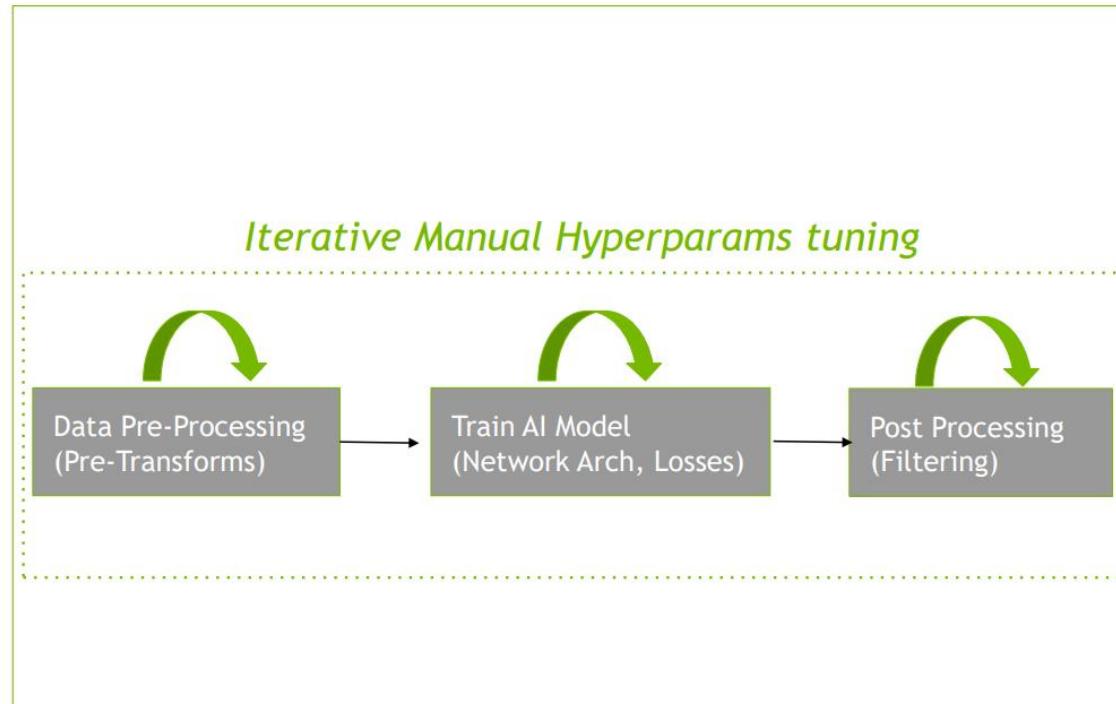
OHIF
AIAA client



MiIL
AIAA client



AutoML



Federated Learning

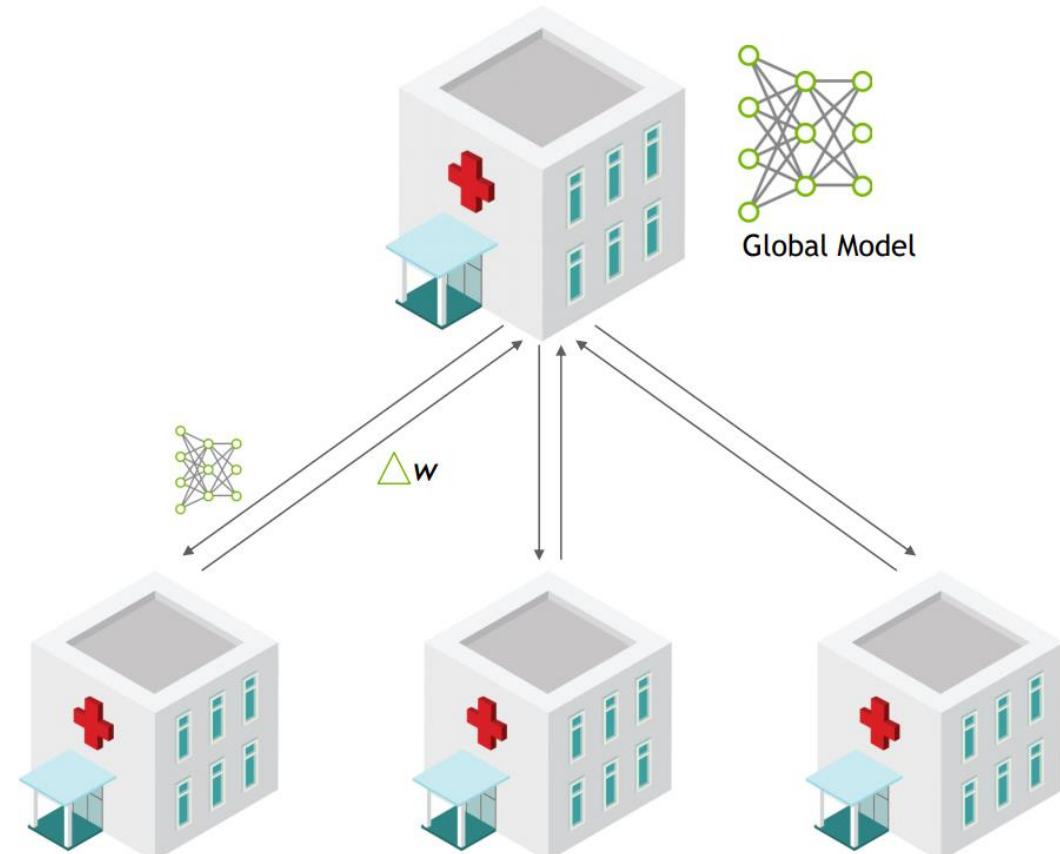
Addressing Data Diversity & Privacy

Distributed Collaborative Learning

Build a common, robust AI model without sharing data

Using NVIDIA Clara to:

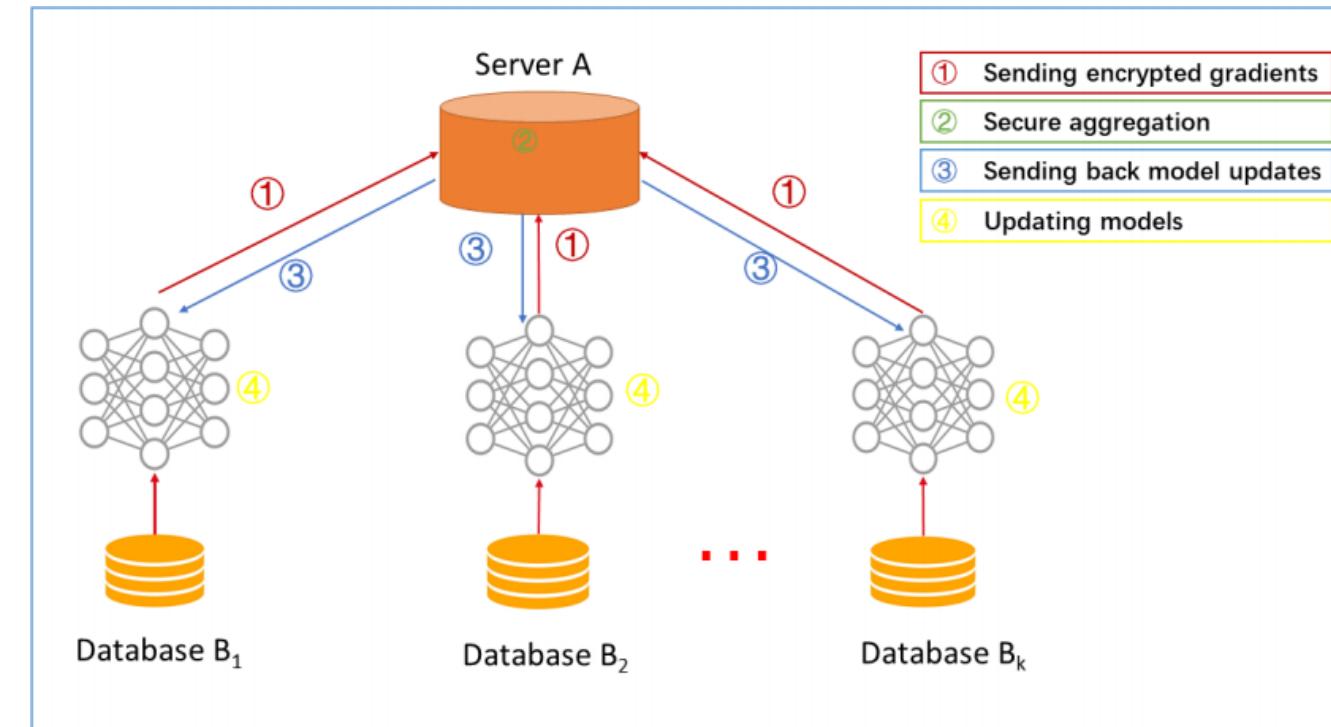
- Authenticate and deliver Clara FL to participating hospitals
- Locally train on private data
- Securely Share partial-model weights
- Apply Federated Averaging creating a new global model
- BYOC to Federated Learning



聯邦式學習 (Federated Learning)

● 2016年Google所提出

- ▶ 為了解決數據孤島問題
 - 符合歐盟GDPR
- ▶ 資料不離開各自的單位
- ▶ 類型
 - 水平式聯邦式學習
 - 垂直式聯邦式學習
 - 聯邦式遷移學習



水平式聯邦式學習

Clara Federated learning

- Server / Client

- ▶ Server load balance not yet
- ▶ Based on HTTPS SSL connection with FL token

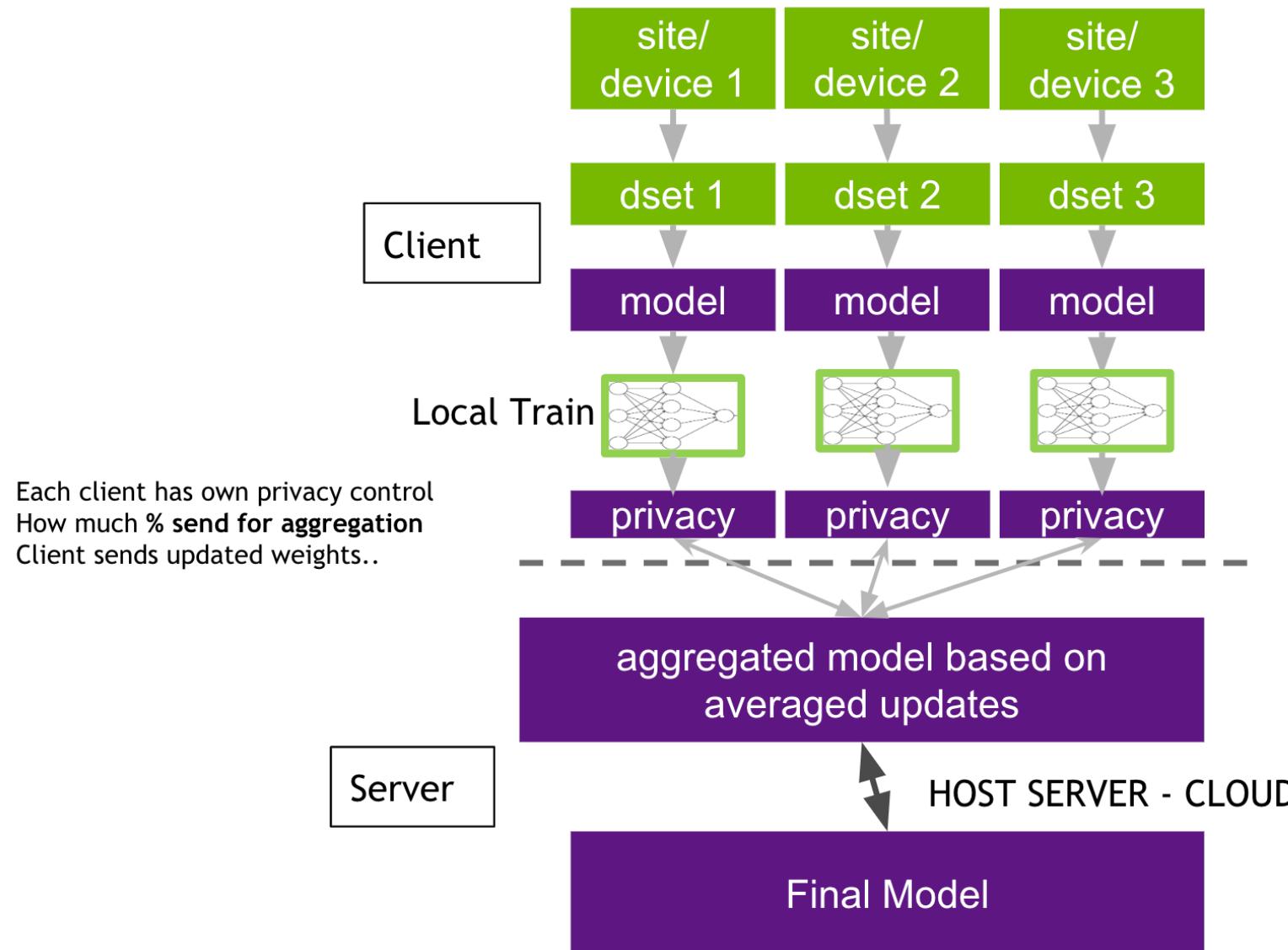
- Packaged with the same MMAR structure

- ▶ Medical Model Archive
 - » a standard structure for organizing all artifacts produced

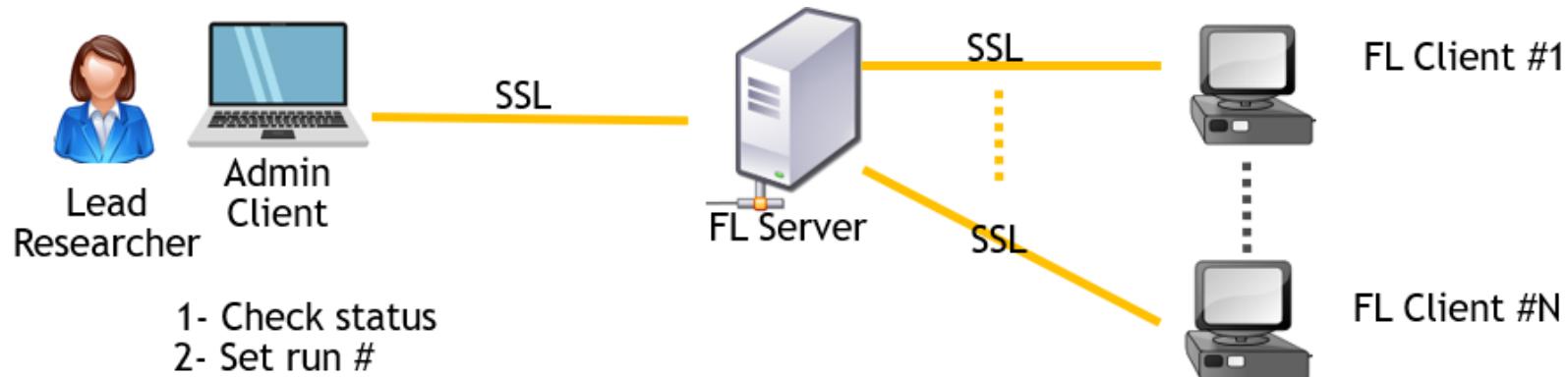
- Targeted users

- ▶ hospitals and medical institutions

Clara Federated Learning Architecture



Clara Federated Learning Workflow



Server Staging



Server Run #

- Clients Run #
- Either for all clients
- or per client

4- Start training



5- Get logs



6- Get metric



7- Shutdown

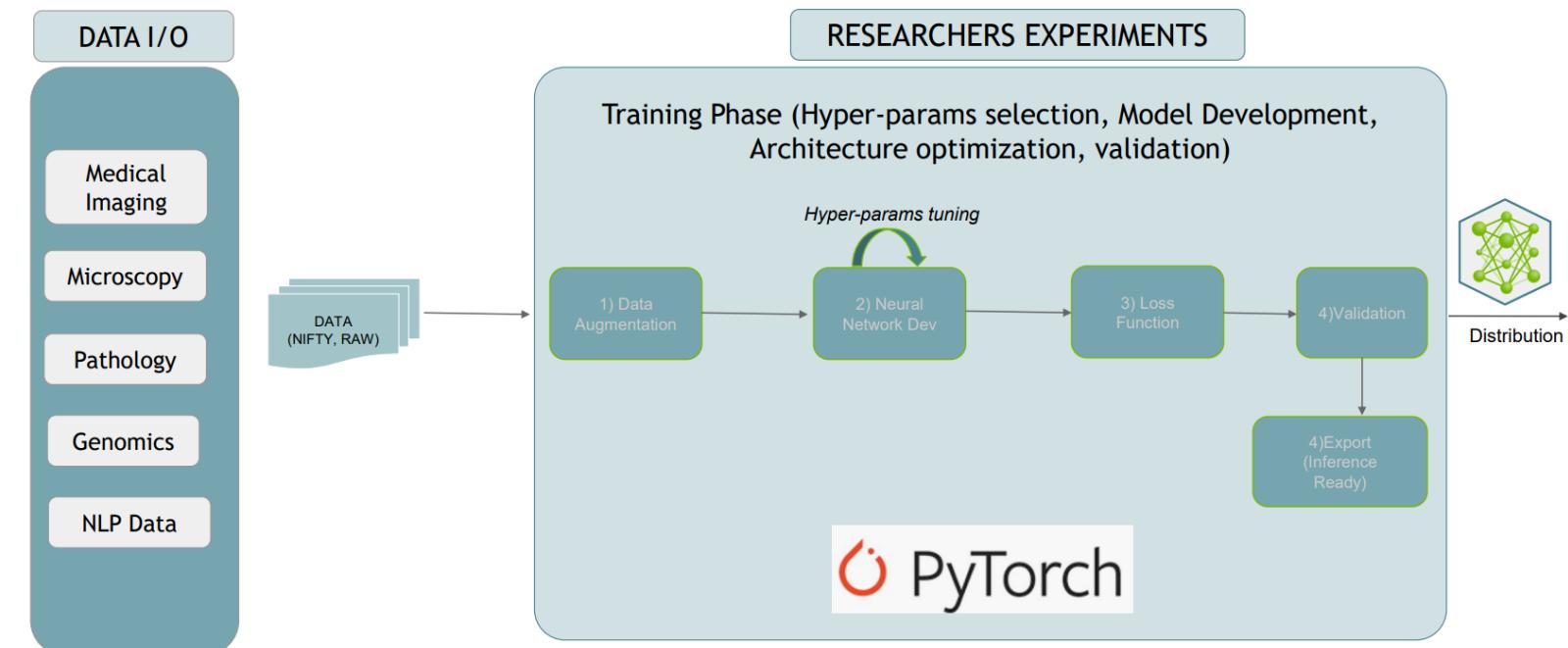


Clara Train SDK 4.0

Clara Train SDK 3.1 至 4.0 差異

○ 整合MONAI Framework

▶ 由TensorFlow轉成MONAI and PyTorch Ignite



Clara Train SDK 4.0 新增功能

● 增加NVFlare 套件

▶ NVFlare (NVIDIA Federated Learning Application Runtime Environment)

- used in contexts outside of Clara as well

- 可抽換掉Clara Train Framework客製化使用其他training framework

● Homomorphic Encryption

▶ Using TenSEAL

- Based on Microsoft SEAL

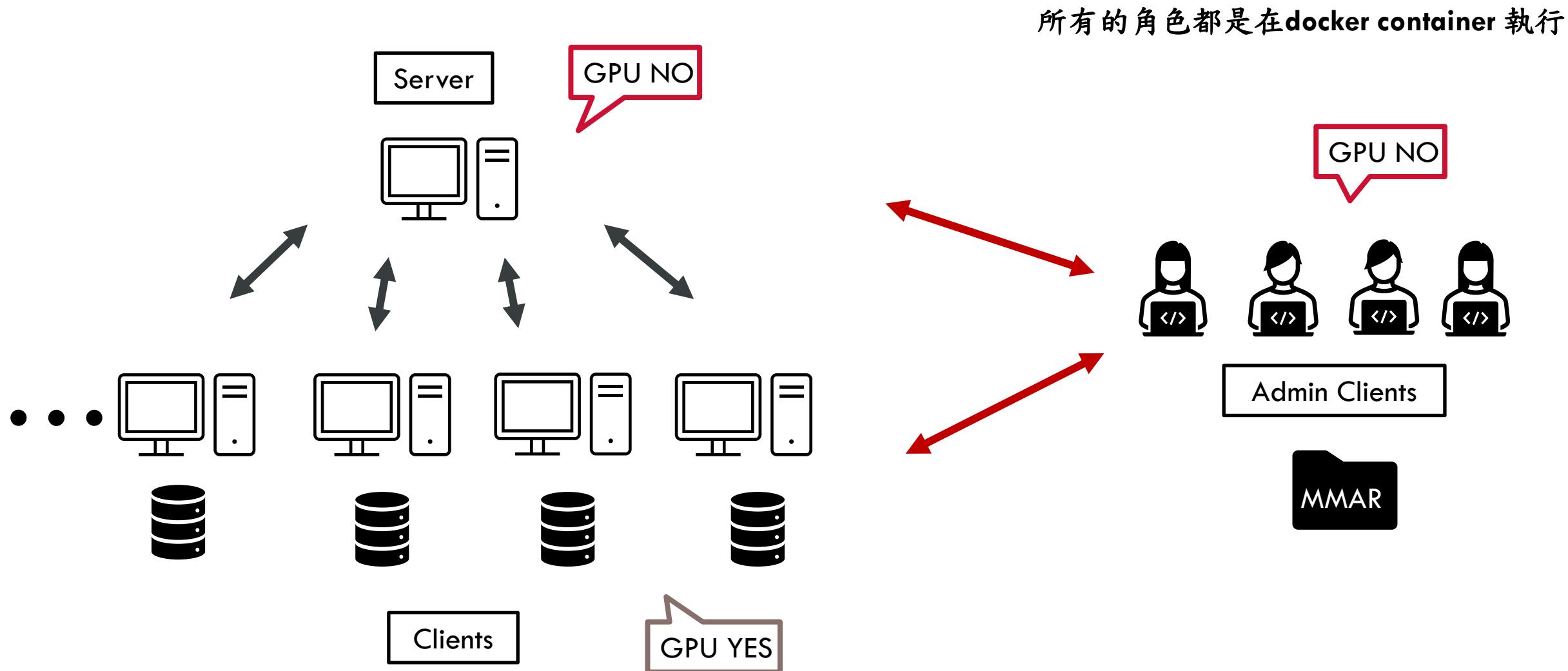
▶ 透過HE的技術讓FL Server不會取得模型參數

- 收集到每個FL Client上傳的模型參數皆已加密

- » 於加密的狀況下進行模型參數聚合

Clara 在 TWCC 的配置流程

Clara Federated Learning 架構配置



Clara Federated Learning環境設定

● Clara軟體在Linux的環境，透過Container運作執行

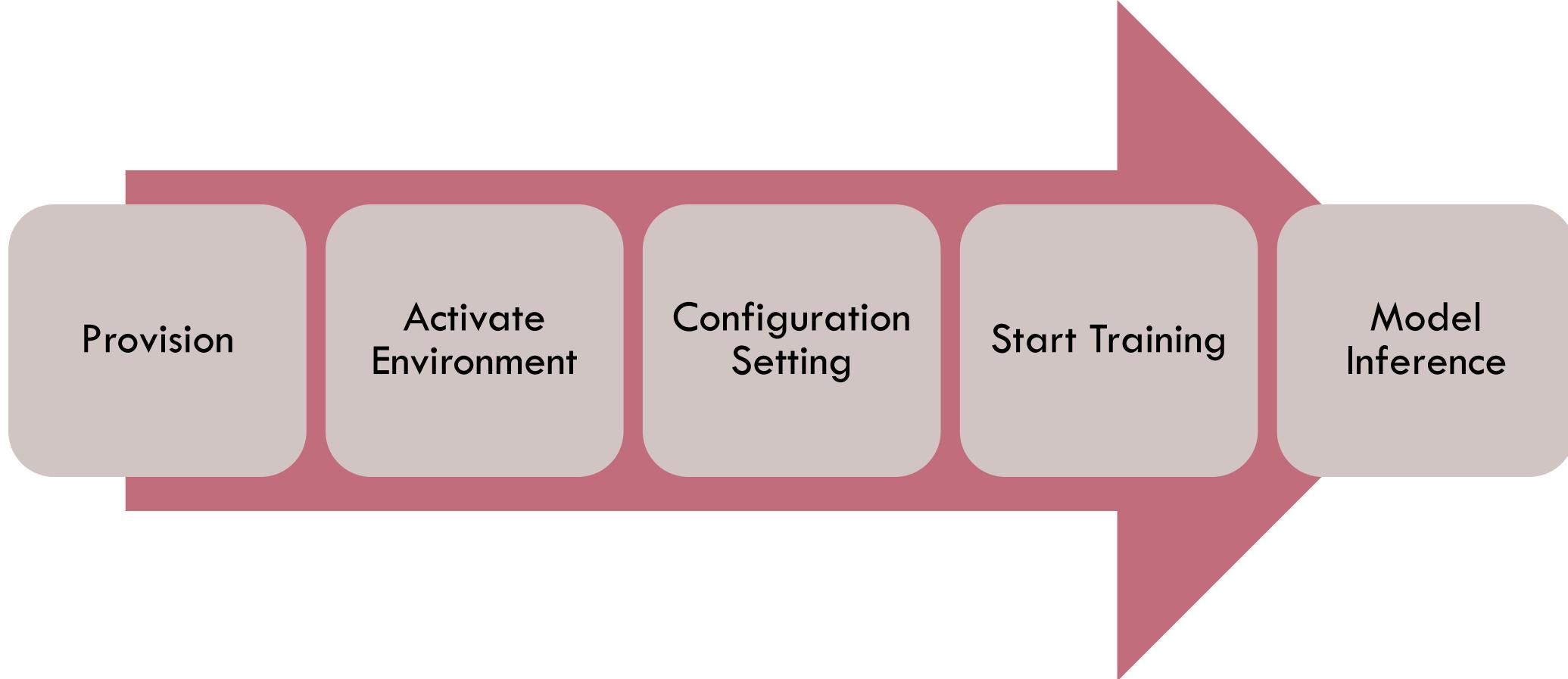
- ▶ 使用TWCC開發型容器開啟
- ▶ 從NGC自行下載配置

● TWCC平台運作配置

- ▶ FL Server
 - 使用TWCC CPU VM
 - 自行配置

- ▶ FL Client
 - 使用TWCC GPU 開發型容器
 - 自行配置

Clara Federated Learning 執行流程



Provision用途

- 產生 Federated Learning 所需要的設定執行檔案

- ▶ 運作程式
- ▶ 連線資訊
- ▶ SSL 加密 key pair

- 加密的 zip 檔案

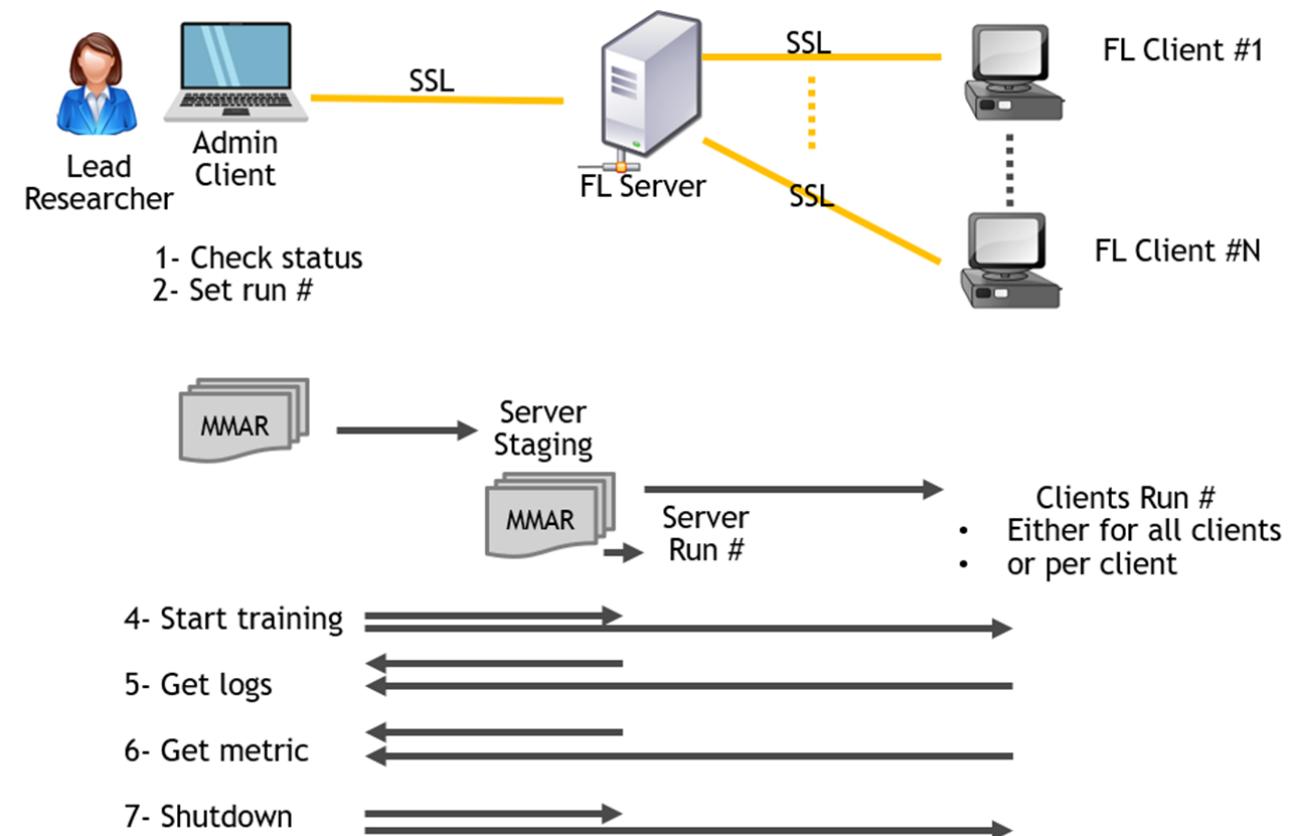
- ▶ FL Server
- ▶ FL Client
- ▶ FL Administration

Provision動作

- 編輯FL設定檔案 project.yml, authz_config.json
 - ▶ FL Server位址
 - ▶ FL Client數目
 - ▶ FL Administration權限
- Provision後將產生的zip放至對應的Container內部
 1. 執行FL Server程式
 2. 執行FL Client程式
 3. 執行FL Administration程式進入CLI介面

Clara 執行流程 - Start Training

- 登入FL Admin 界面
 - ▶ 操作CLI指令



Clara 執行流程 - Model Inference

● Do yourself

- ▶ Pickup model at FL client side
 - Models in MMAR folder

● Using AIAA in Clara Container

- ▶ Inference through restful API

Clara 測試

測試環境

- 使用 Clara 3.1
- 使用 TWCC VM 作為 FL Server
 - ▶ 1 VM
- 使用 TWCC V100 container 作為 FL Client
 - ▶ 1~8 Container

測試資料

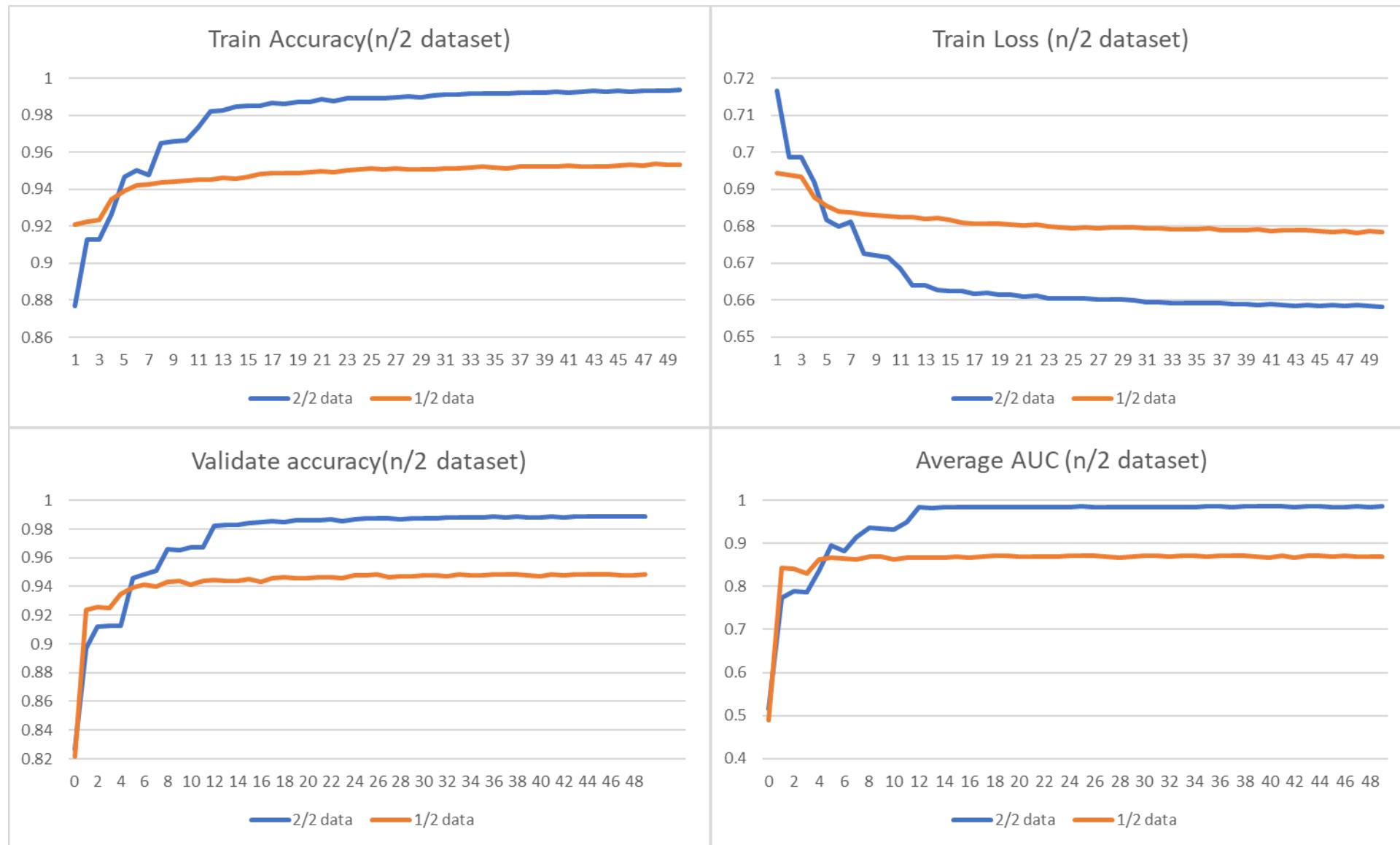
- 資料集: MNIST (<http://yann.lecun.com/exdb/mnist/>)
 - ▶ 60000 28x28 training image
 - ▶ 10000 28x28 validation image
- 模型: *# Model Structure*

```
model = Sequential()
model.add(Conv2D(filters=32, kernel_size=3, input_shape=(1, 28, 28),
activation='relu', padding='same'))
model.add(MaxPool2D(pool_size=2, data_format='channels_first'))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(10, activation='softmax'))
```
- Loss function: crossentropy, Optimizer: Adam, Metrics: Accuracy
- Parties: 2, 4, 8,
- fusion freq. = 1, 2, 5

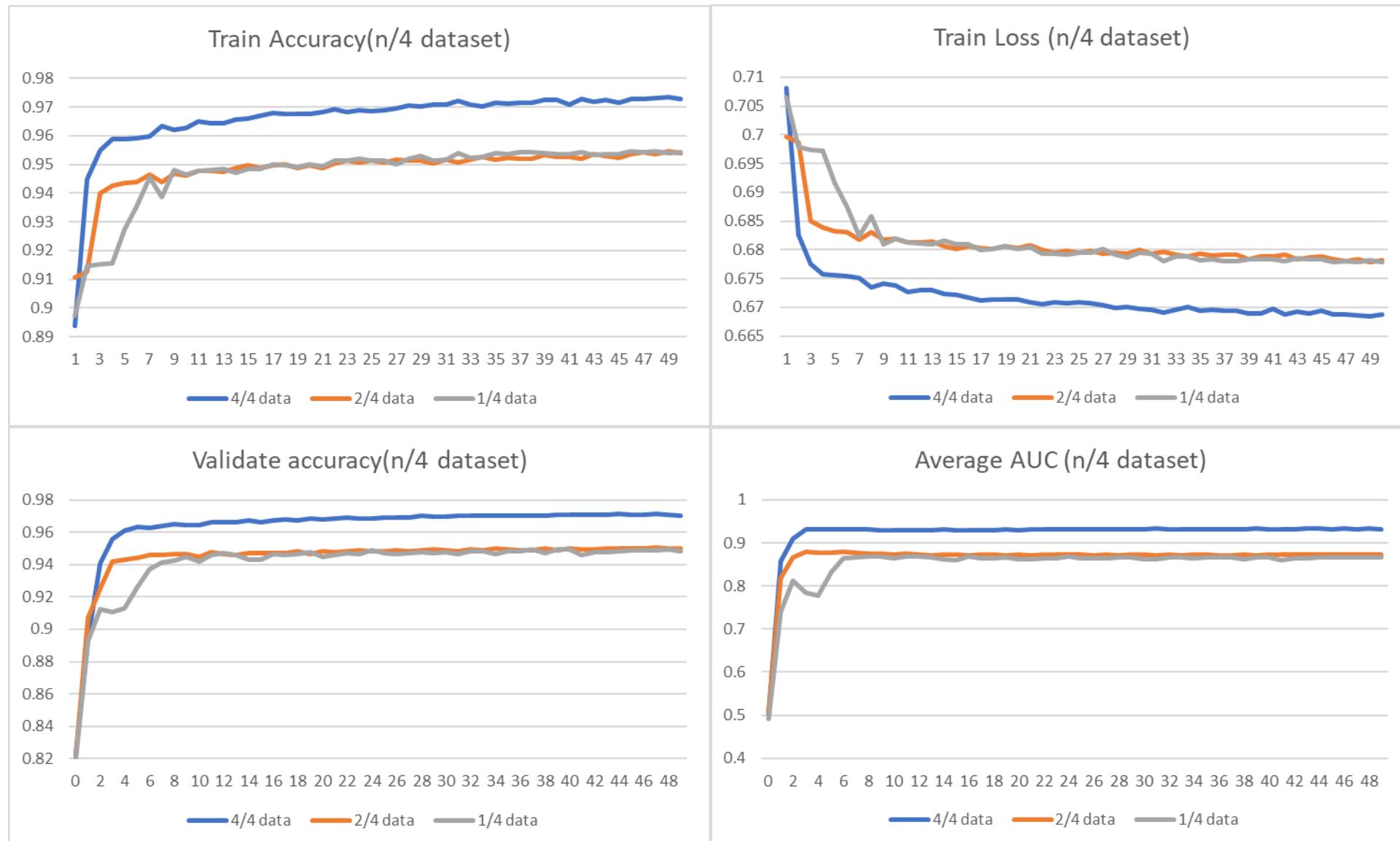
於2/4/8個FL Clients時disable部份FL Clients
只用部份比例資料進行訓練

Clara不同資料數目的訓練結果

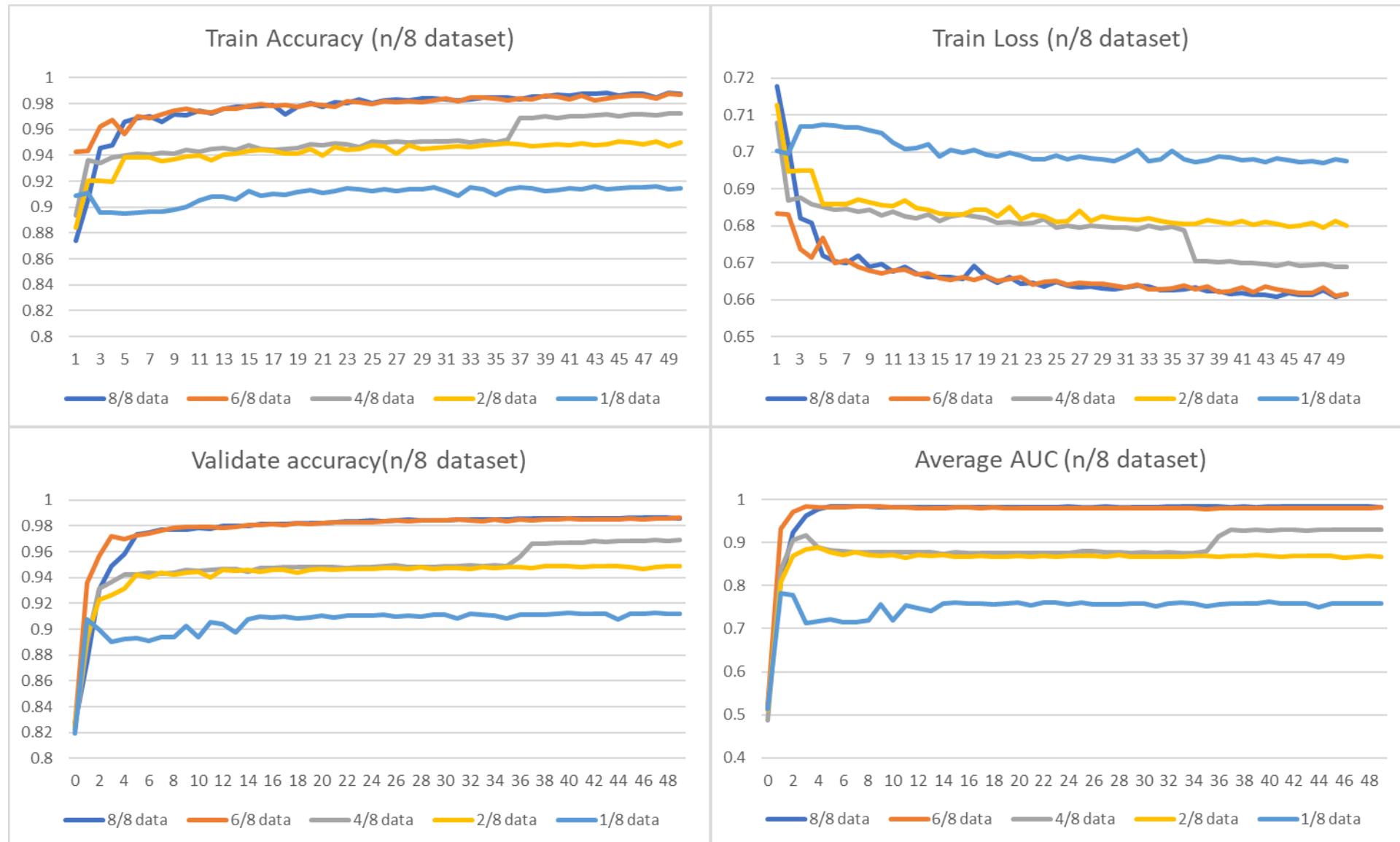
測試結果:2 Clients,2 Epoch/Round



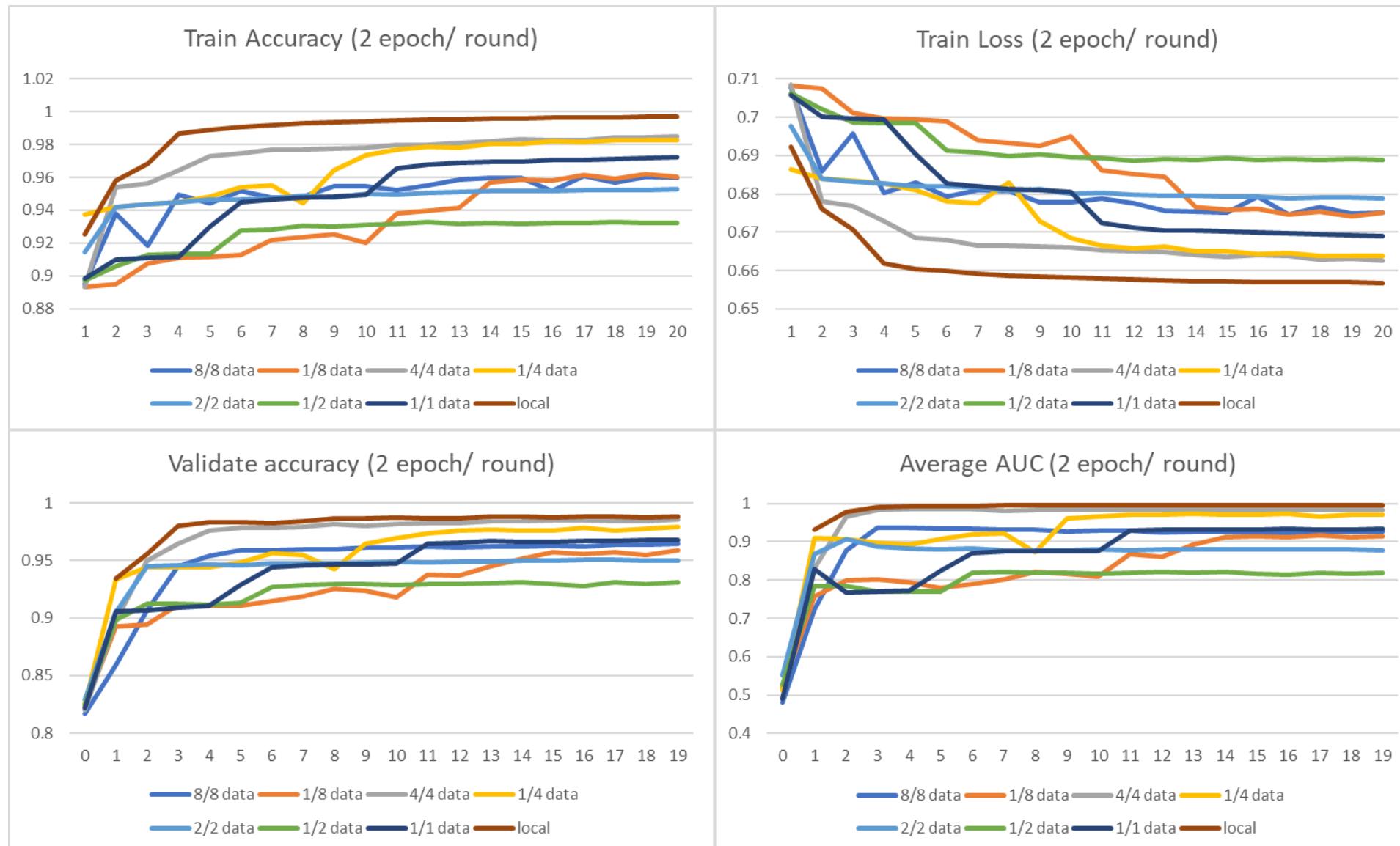
測試結果:4 Clients,2Epoch/Round



測試結果:8 Clients, 2 Epoch/Round



測試結果: 2 Epoch/Round



分為1/2/4/8個FL Client

不同FL Client數目的訓練結果

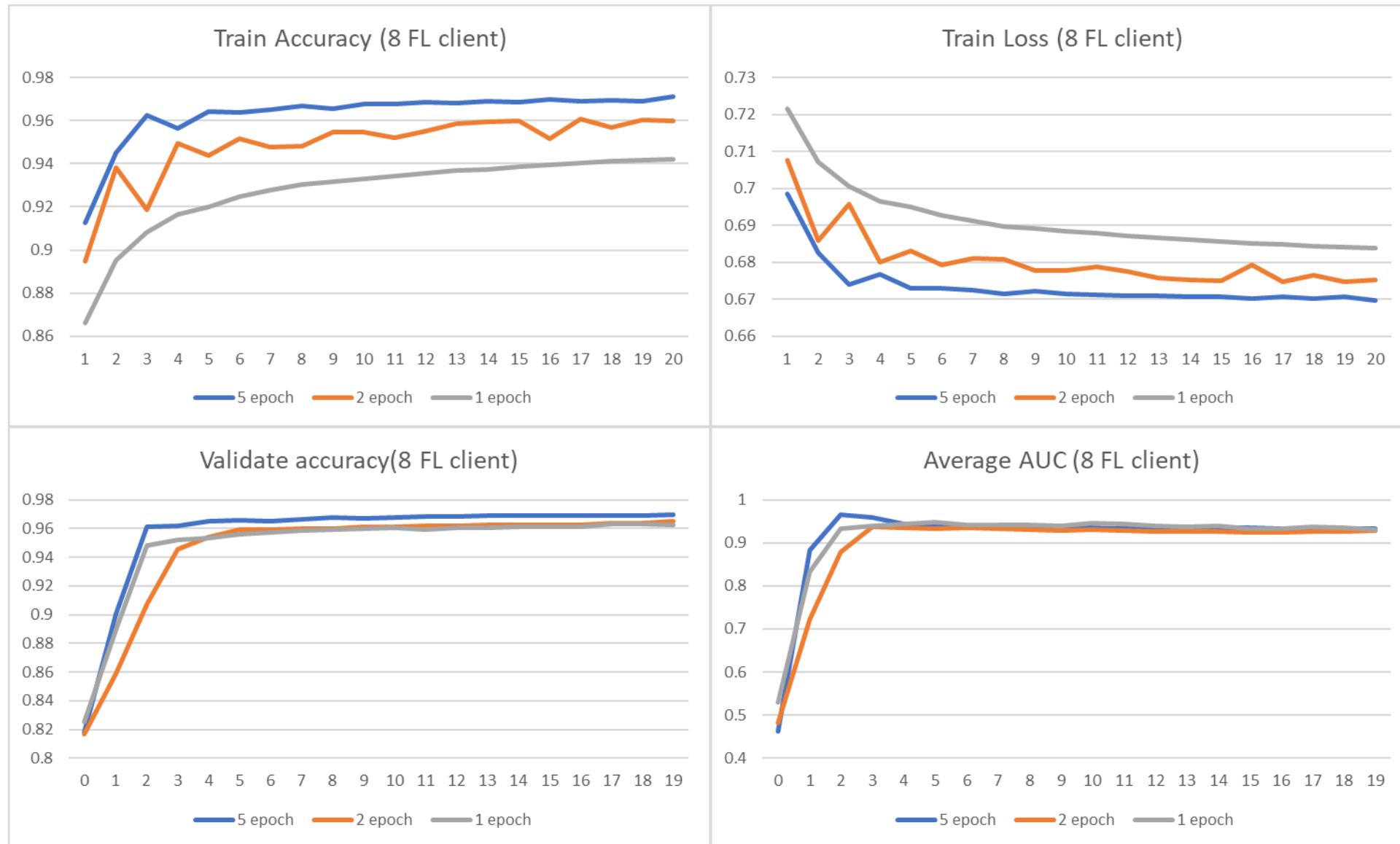
測試結果: 2 Epoch / Round



分為1/2/5 Epoch / Round的Aggregation rate

不同Aggregation rate的訓練結果

測試結果: 8 Parties



Clara在NCHC的應用

Clara在NCHC的應用現狀-1

● Provide Nvidia Clara Train SDK Container on TWCC

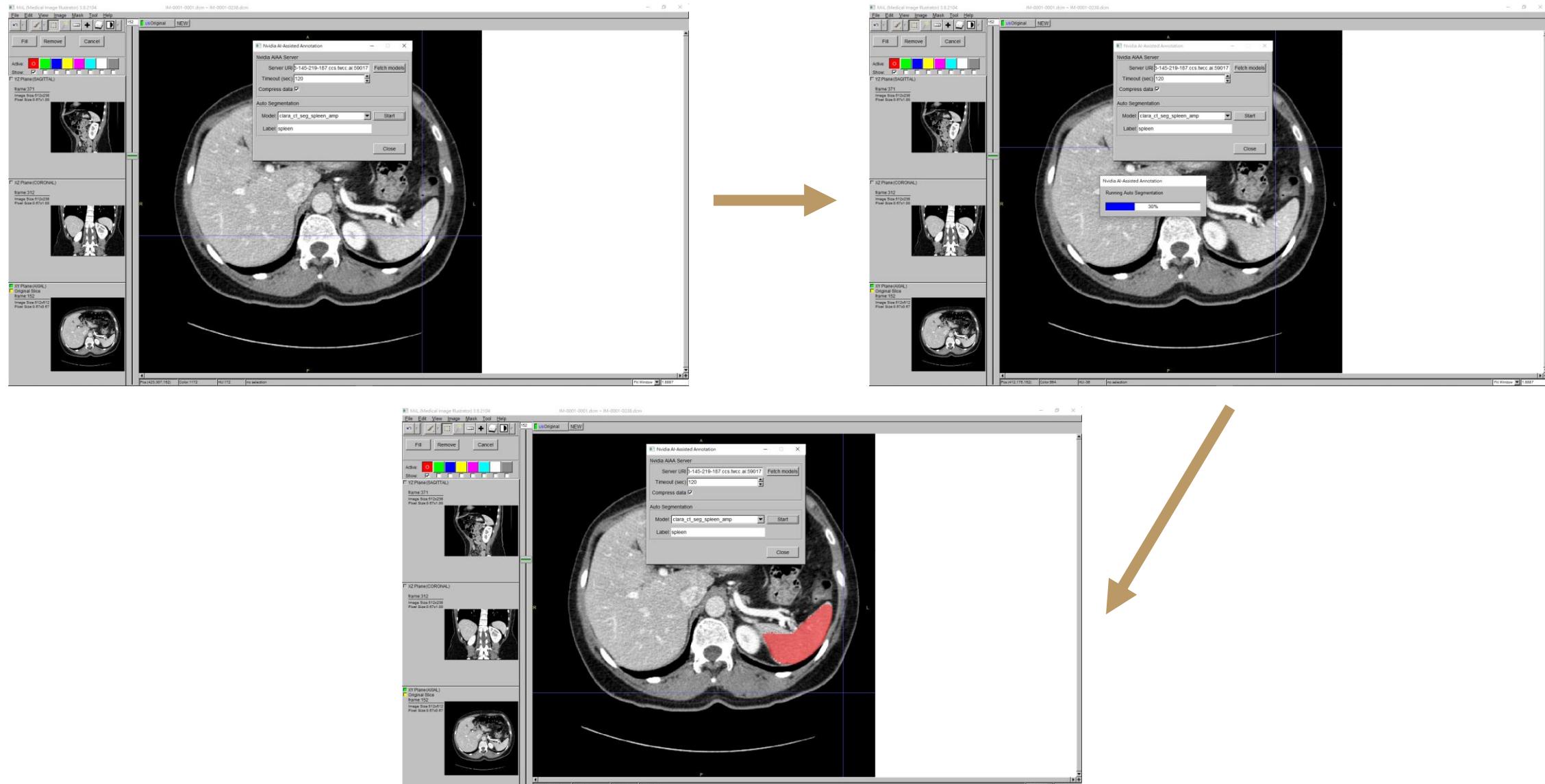
- ▶ 現有3.1版，4.0版已申請上架中
- ▶ 直接使用開發型容器開啟
- ▶ 與廠商合作測試使用Clara FL 訓練真實資料

Clara在NCHC的應用現狀-2

● 使用MiiL + Clara AIAA功能

- ▶ MiiL: NCHC開發的影像切割軟體
 - ▶ AIAA可提供使用FL訓練完成之模型進行推論
- 使用Clara Train SDK Container 與 MiiL
- » 應用MiiL進行標記準備訓練
 - » 應用FL訓練結果讓MiiL 標記驗證
 - » Training and validation forever

MiiL + AIAA



Clara在NCHC的應用現狀-3

- Provide Nvidia Clara Web UI

- ▶ 降低操作與設定門檻
- ▶ 整合至現有Clara 3.1 Container內

Nvidia Clara Web UI

CLARA FL STEPS

Provision

Steps:

- 修改框后的 "name"，输入本次训练的客户名称。
- 修改其他 Config UI 内的设置。（遵循：依训练所需而定。）

Download

Project: dalin_project2

Main

name: dalin_project2

Panel of Main

Server

org: nchc
ip: 203.145.222.103.twcc.ai
fed.learn_port: 443
admin_port: 80
admin_storage: transfer
min_num_clients: 1
max_num_clients: 100

Panel of server

admin_clients

org: nchc
email: xjapan666@nchc.narl.org.tw

Panel of admin_clients



CLARA FL STEPS

Download Packages

Steps: (以下步驟皆於電腦上執行)

- 下載後方Cmd選項(黑底白字)，輸入：“./startup/fl_admin.sh”
- 登入Admin tool，輸入：“帳號”(您的登入email)
- Check status: (可略過)
 - 輸入：“check_status server”(檢查 server 狀態)
 - 輸入：“check_status client”(檢查 client 狀態)
- 輸入：“set_run_number 1”(輸入次訓練的 Run Number(ex. '1')，與與下方 Run Number 的 input 逐一設)

Next

Panel of fl_clients

Type	Filename	Password
client	fl_site1.zip	NtzXb0QgD075ZEpA
client	fl_site2.zip	cJxGrWfAiC3uzqX1



CLARA FL STEPS

Web-Admin-Tool

Steps:

- 上傳 mmar: 請將 mmar 資料夾 zip 壓縮上傳。
- 將 mmar 外掛 Folder -> zip
- 總體壓縮大小約 < 1G
- 若此 project 已上傳 mmar 可直接此步驟。
(至 /opt/dalindcal/tools/Project_Name/Admin_ID/transfer)
- 輸入：“upload_folder MMAR_Folder” (MMAR_Folder: 當的 MMAR 資料夾名稱。ex.“clara_dalin_heart_light.zip”)
- 輸入：“upload_folder MMAR_Folder” (MMAR_Folder: 當的 MMAR 資料夾名稱。ex.“clara_dalin_heart”)

Next

jupyter

```

www-data@vinclaserfl-1699128-lease:/opt/nvidia/medice/1/tools/dalin_project2/xjspan6
669nchc.narl.org.tw:22 2021-06-30 13:39:16.949603
User Name: xjpan666@nchc.narl.org.tw
Type ? to list commands; type ? cmdName? to show usage of a command.
FL run number has not been set.
FL server status: training act started
Registered clients: 1
| CLIENT NAME | TOKEN
| fl_site1 | 810719a9-a4ee-47e9-9970-b25ea5134293 |
| fl_site2 | 2021-06-30 13:39:16.949603
| Net run number:
run number already exist. Set the FL run number to 1.
Done [199ms] 2021-06-30 13:39:15.953304
uploaded folder: /opt/nvidia/medice/1/tools/dalin_project2/server/startup/../.transfer
clara_dalin_heart
Done [62160 us] 2021-06-30 13:40:19.907603

```

Implement by Jupyter notebook

fl_site2.zip

fl_site1.zip

Nvidia Clara Web UI

CLARA FL STEPS

Server Setting

Steps:

- 修改适合于你的 Config UI 内的设置。(谨慎，未经授权的配置可能造成损害)
- 需要执行 "Apply" 将上面修改保存，并部署。
- 输入 "deploy MMAR Folder server" (该 mmar 必要部分部署至 server)
- MMAR Folder 目录的 MMAR 文件夹名。(ex: "clara_dalin_heart")
- 上传 Datafile.json。(注: _run_Xmmar_server(必须先执行上一步才可上传))

datafile.json

完成后将返回：完成所有必要的准备工作后，请点击“Next”跳至下一步。

jupyter

```
Administrator: 203.145.222.103:60001 on port 80
User Name: xjpan466@chu.narl.org.tw
Type ? to list commands; type "cmdName" to show usage of a command.
? check_jupyter_server
No Jupyter server has been set.
$FL server status: training not started
Registered clients: 1

| CLIENT NAME | TOKEN | SUBMITTED MODEL |
| f1_site1 | 810719e9-a6ea-47f9-9970-b29e0134293 |



Done [590 usecs] 2021-06-30 13:59:46.696803
run number already exist. Set the FL run number to 1.
Done [599 usecs] 2021-06-30 13:59:51.953304
Done [599 usecs] 2021-06-30 13:59:51.953304
up /opt/nvidia/medina/dalin_project2/...
Created folder /opt/nvidia/medina/tools/dalin_project2/server/startup/.../transfer
/dalin/hear...
Done [452 usecs] 2021-06-30 13:40:13.907603
> deploy clara_dalin_heart
server has been deployed.
Done [10469 usecs] 2021-06-30 13:41:42.758596
```

implement by jupyter notebook



CLARA FL STEPS

Clients Setting

Steps:

- 修改适合于你的 DATA_ROOT，插入一些训练数据的临时路径。(ex: /home/u6003490/Data/SegmentRegions)
- 修改适合于你的 datasetlist.json 的临时路径。(ex: /home/u6003490/Data/datasetlist.json)
- 修改其他 Config UI 内的设置。(谨慎，未经授权的配置可能造成损害)
- 需要执行 "Apply"，跳上操作的步骤才能部署。
- 输入 "deploy MMAR Folder client Your_Site_Name" (该 mmar 必要部分部署至 client)
- MMAR Folder 目录的 MMAR 文件夹名。(ex: "clara_dalin_heart")
- Your_Site_Name: 你的部署别名。(子壳).zip, ex: "clara_dalin_heart"
- 输入你的部署别名。(ex: "f1_site1")

datafile.json

完成后将返回：完成所有必要的准备工作后，请点击“Next”跳至下一步。

jupyter

```
FL server status: training not started
Registered clients: 1

| CLIENT NAME | TOKEN | SUBMITTED MODEL |
| f1_site1 | 810719e9-a6ea-47f9-9970-b29e0134293 |



Done [820 usecs] 2021-06-30 13:59:36.696803
run number already exist. Set the FL run number to 1.
None [5990 usecs] 2021-06-30 13:59:51.953304
up /opt/nvidia/medina/dalin_project2/...
Created folder /opt/nvidia/medina/tools/dalin_project2/server/startup/.../transfer
/dalin/hear...
Done [452 usecs] 2021-06-30 13:40:19.007603
> deploy clara_dalin_heart server
server has been deployed.
Done [10469 usecs] 2021-06-30 13:41:42.758596
> deploy clara_dalin_heart client
instancef1_site1 : MMAR deployed.
instancef1_site1 : MMAR deployed.
Done [536067 usecs] 2021-06-30 13:43:44.879295
```

implement by jupyter notebook



CLARA FL STEPS

Training

Steps:

- 输入 "start server" (Server 准备好后 Client 可以使用。)
- 输入 "start client" (Client 启动训练。)

FL 共分七步骤，完成每一步骤之后请逐步依次操作至左下方，最后一步骤。

jupyter

```
None [65337 ussecs] 2021-06-30 13:59:36.696803
> set_run_number 1
run number already exist. Set the run number to 1.
None [5990 usecs] 2021-06-30 13:59:51.953304
> upload folder clara_dalin_heart
Created folder /opt/nvidia/medina/tools/dalin_project2/server/startup/.../transfer
/dalin/hear...
Done [452 usecs] 2021-06-30 13:40:19.007603
> start server
Server training is starting...
Done [506470 ussecs] 2021-06-30 13:45:01.739789
> start client
instancef1_site1 : start the client...
instancef1_site2 : start the client...
Done [506470 ussecs] 2021-06-30 13:45:15.249256
> [ ]
```

implement by jupyter notebook

Env Train Validation Validation ckpt Cross Site Validation Server Client

servers

name:	Dalin_heart_test
service:	

client

- local_epochs: 2
- steps_aggregation: 0
- exclude_vars: dummy
- cross_site_validate: 1
- my_rank: 0
- privacy: name

environment.json

```
DATA_ROOT: /home/u6003490/Data/SegmentRegions
DATASET_JSON: /home/u6003490/Data/datasetlist.json
PROCESSING_TASK: classification
MMAR_EVAL_OUTPUT_PATH: eval
MMAR_CKPT_DIR: models
PRETRAIN_WEIGHTS_FILE: 0
```

Nvidia Clara Web UI

● FL Server / FL Client / FL Administration

- ▶ 皆透過登入Web頁面進行不同權限操作
- ▶ Web UI提供參數設定與操作
- ▶ 使用資料庫紀錄訓練參數資訊
 - 包含FL Client各自客製化自己的環境參數
- ▶ 預計一個月後完成

The screenshot displays three main sections of the Nvidia Clara Web UI:

- CLARA FL STEPS:** A dashboard showing the status of training steps. It includes a "Result" section with a list of actions:
 - 1. 命令: "abort client" (強制結束 client 之訓練 - 可取消。)
 - 2. 命令: "abort server" (手動結束 server - 中止之 client 也将一律停止。)
 - 3. 命令: "shutdown client" (關閉 client 程序 - 點輸入 email -)
 - 4. 命令: "shutdown server" (關閉本次 FL Server 程序 - 點輸入 email -)
 A "Stop" button is also present.
- jupyter:** A terminal window showing the logs of a Jupyter notebook execution. The logs include commands like "start server", "deploy clara.dain.heart.server", and "stop server".
- 成果檔案列表:** A table listing files categorized by client and validation type:

Clients/CrossValid	Filename
fl_site1	modelckpt.data-00000-of-00001 modelckpt.index modelckpt.meta
fl_site2	modelckpt.data-00000-of-00001 modelckpt.index modelckpt.meta
Cross Validation	cross_val_results.json

Clara在NCHC的應用 -4

● 研華AIoT

- ▶ 於AIFS portal提供Clara 3.0 版FL功能

● Asus AI maker

- ▶ 整合Nvidia Clara 並預計加入Clara FL功能
- ▶ 暫定2021年底提供服務

感想

○ 從Nvidia Digits到Clara Train SDK

▶ 對DL應用的方向設計

- 包含了inference應用的AIAA
- 其他Clara Deploy Container可整合
- 透過內建 module與設定檔降低programming需求
- 提供資料前處理的功能

○ MMAR的版本有點混亂

▶ 4.0版開始修正這個問題

○ Json與yml設定檔在User Guide的描述

Thanks for Listening.

Contact: bylee@narlabs.org.tw