SUPPLEMENTARY MATERIAL

Reinforced Labels: Multi-Agent Deep Reinforcement Learning for Point-Feature Label Placement

1 TRAINING & HYPERPARAMETERS

We use PPO implementation from the RLLib framework [1]. Rollout workers query the current policy to determine actions and collect a new vector of observations and rewards. Collected data are assembled into training batches and shuffled. The trainer worker coordinates the rollout workers and orchestrates policy optimization. We use Adam to optimize the policy and value function parameters. Further hyperparameters are as follows:

TABLE 1
Hyperparameters

Parameter	Value
Learning rate	$1 \cdot 10^{-6}$
Batch size	2000
Mini-batch size	128
SGD epochs	10
PPO clip factor	0.25
PPO gradient clipping	–
PPO entropy coefficient	0.0
Horizon	100
γ	0.99
λ	1.0

2 OBSERVATION MODALITIES

We designed two observation vectors. First, the *mapping* vector M provides surroundings modalities. Second, the *self-aware* vector S provides local information about the state of the given label agent. Observations are mostly normalized to $[-1,1]$ range. The intersection observation type is an exception – the value of -1 signifies the intersection of anchor, 0 defines screen bound, and 1 represents the label. We use Box2D[1] framework to cast rays serving as LiDAR sensor to encode the agent's surroundings by 32 rays (similar to LiDAR) evenly distributed around the label bounds that sense the range, type of the nearest intersected object (*i.e.*, label, anchor, bounds of the environment), and the mass that the ray went through.

3 DATASET DEFINITION

The dataset comprises a collection of instances, where a JSON file represents each instance. The JSON file contains information about the screen or drawing area, including the width and height, as well as an optional reference to a background image. Additionally, the JSON file contains a description of the labels associated with the instance, where each label is characterized by the position of its anchor and the size of its label box. Moreover, the label definition may include a text string and font configuration for each label.

TABLE 2
Observation Modalities

Mapping vector	Range	Shape
Intersection distance	$(-1,1)$	32
Intersection mass	$(-1,1)$	32
Number of intersected labels	$(-1,1)$	32
Intersection object type	$(-1,0,1)$	32
Self-aware vector	Range	Shape
Overlap area	$(-1,1)$	1
Overlap indicator	$(-1,1)$	1
Number of overlaps	$(-1,1)$	1
Displacement	$(-1,1)$	1
Cumulative displacement	$(-1,1)$	1
Anchor penetration distance	$(-1,1)$	1
Penetration indicator	$(-1,1)$	1
Number of penetrations	$(-1,1)$	1
Anchor-port distance	$(-1,1)$	1
Anchor-port angle	$(-1,1)$	1
Anchor-port vector	$(-1,1)$	2
Anchor-origin distance	$(-1,1)$	1
Elapsed time steps	$(-1,1)$	1

```
{
    "screen": {
        "size": {"width": 2400, "height": 1600},
        "background": {"file": "background.svg"}
    },
    "labels": {
        "definition": {
            "0": {
                "text": "Brno",
                "size": [76, 20],
                "font": {
                    "size": 17,
                    "family": "Arial",
                    "style": "Regular"}
            },
            ...
}
```

REFERENCES

[1] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. Gonzalez, M. I. Jordan, and I. Stoica, "Rllib: Abstractions for distributed reinforcement learning," in *Proceedings of the 35th ICML 2018*. PMLR, 2018, pp. 3059–3068.

1. The framework Box2D is available at https://box2d.org.

TABLE 3
Visual comparison of examined methods applied to selected instances from the compact dataset and real-world instances of IATA airport codes with 250 anchors and CITY names with 150 anchors based on data obtained from Open Street Maps. The green dot represents the anchor (*i.e.*, point feature). The gray rectangle symbolizes the body of the label itself. The red dot describes an anchor that was not labeled by the given method. The red rectangle illustrates the dimensions of the missing label. We stress that we intentionally added all the missing labels to the visualization for illustrative purposes only, and their origins are not the outcome of the method itself.

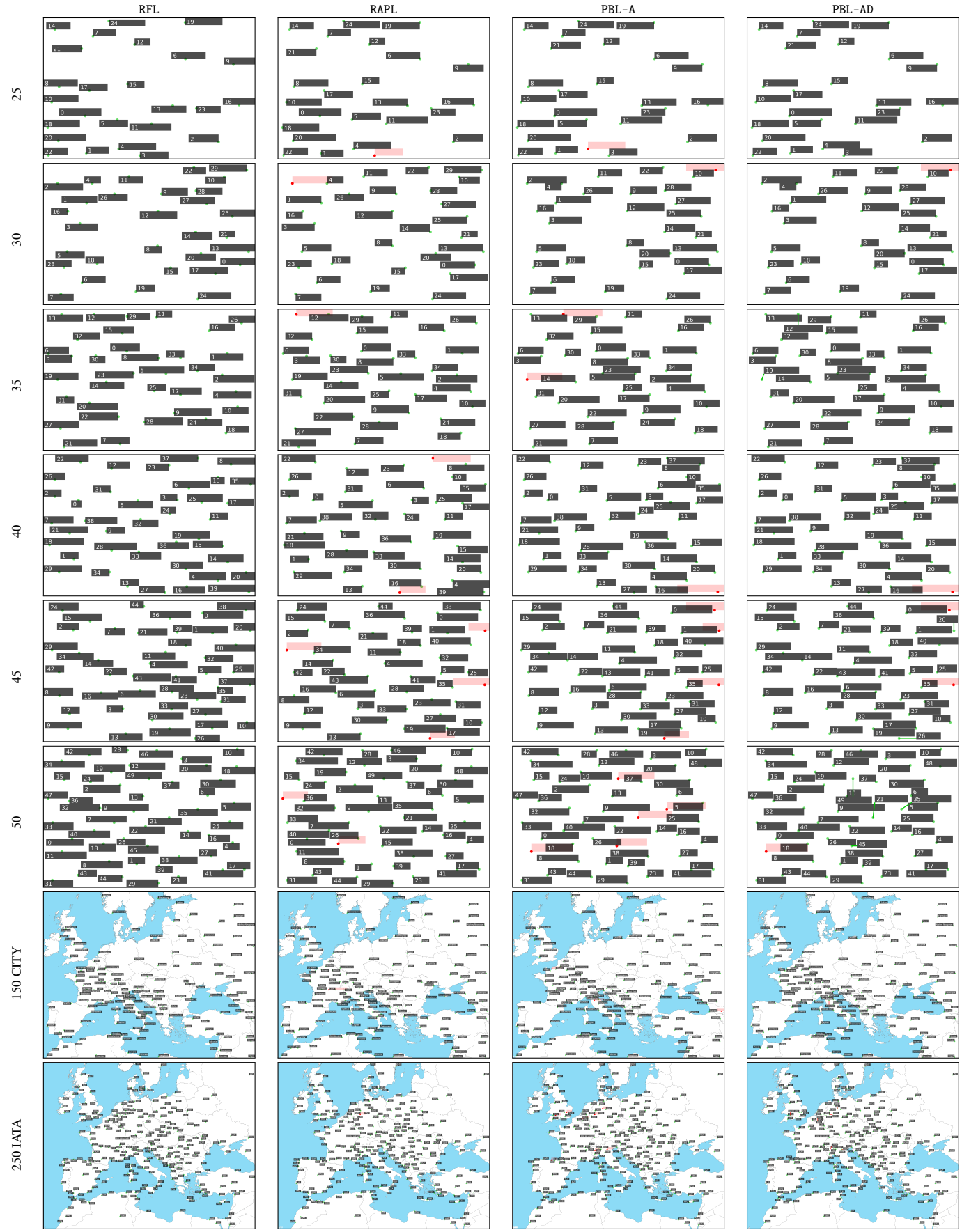


TABLE 4
Visual comparison of RFL and RAPL methods applied to selected instances from the volume dataset.

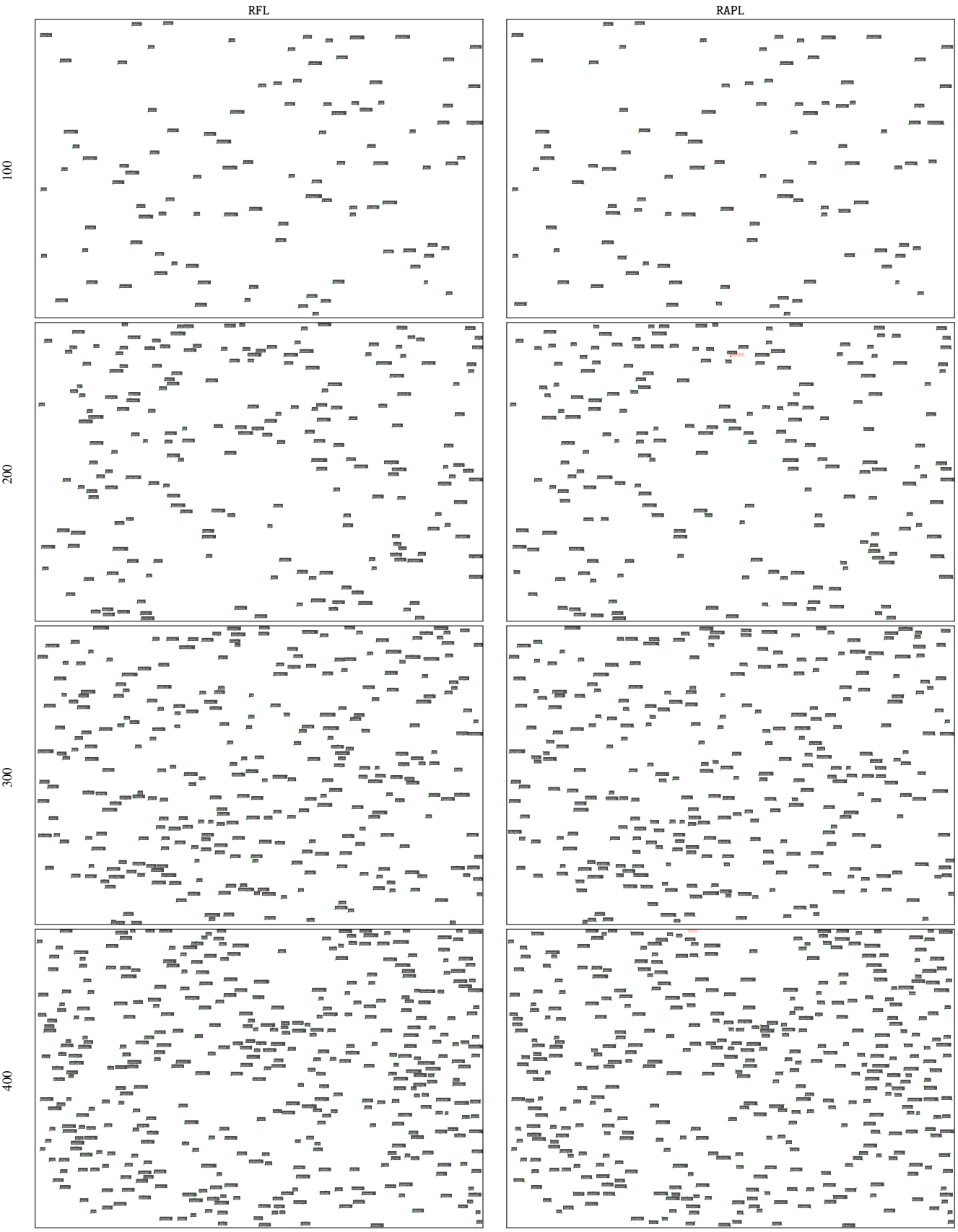


TABLE 5
Visual comparison of RFL and RAPL methods applied to selected instances from the volume dataset.

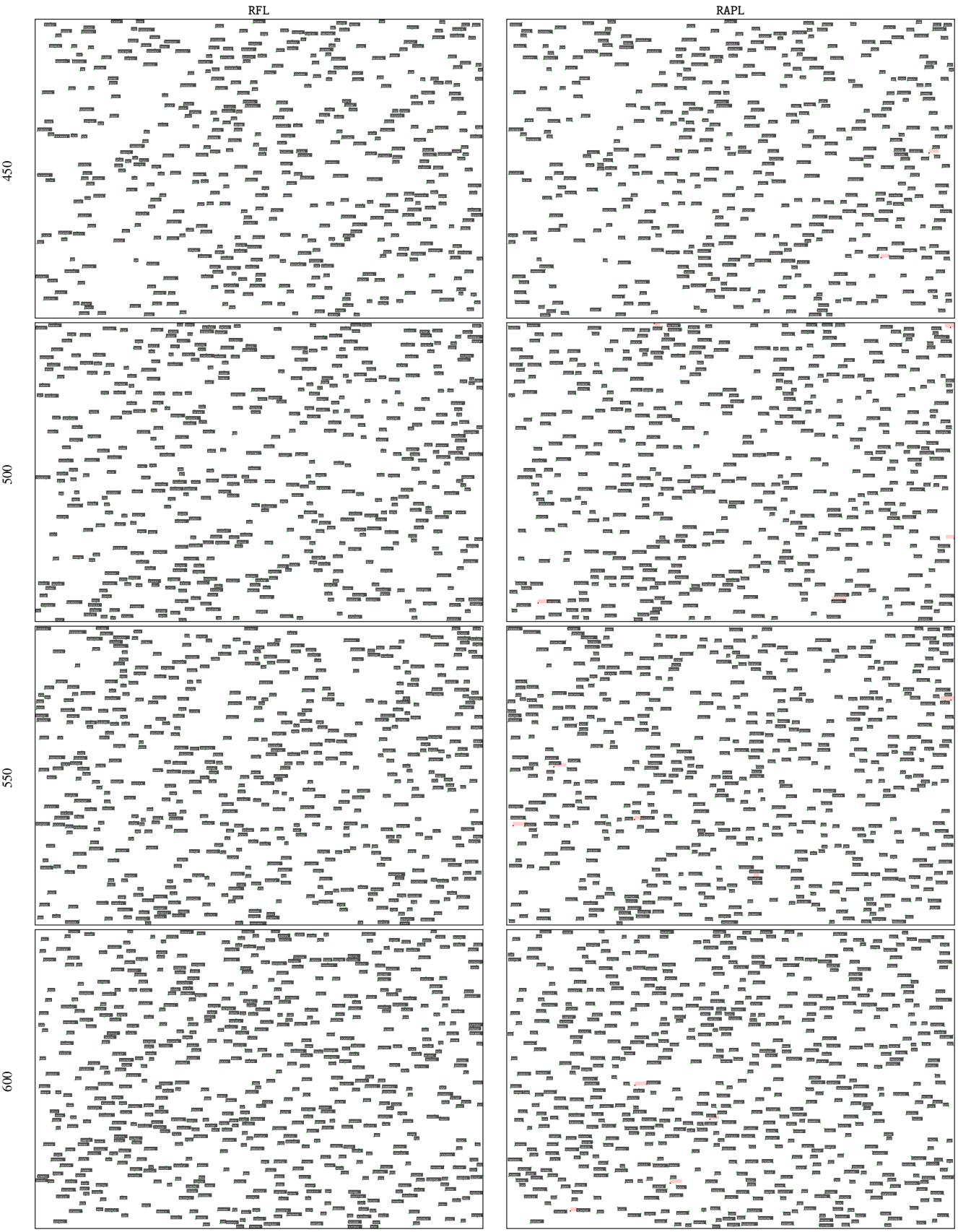


TABLE 6
Visual comparison of PBL-A and PBL-AD methods applied to selected instances from the volume dataset.

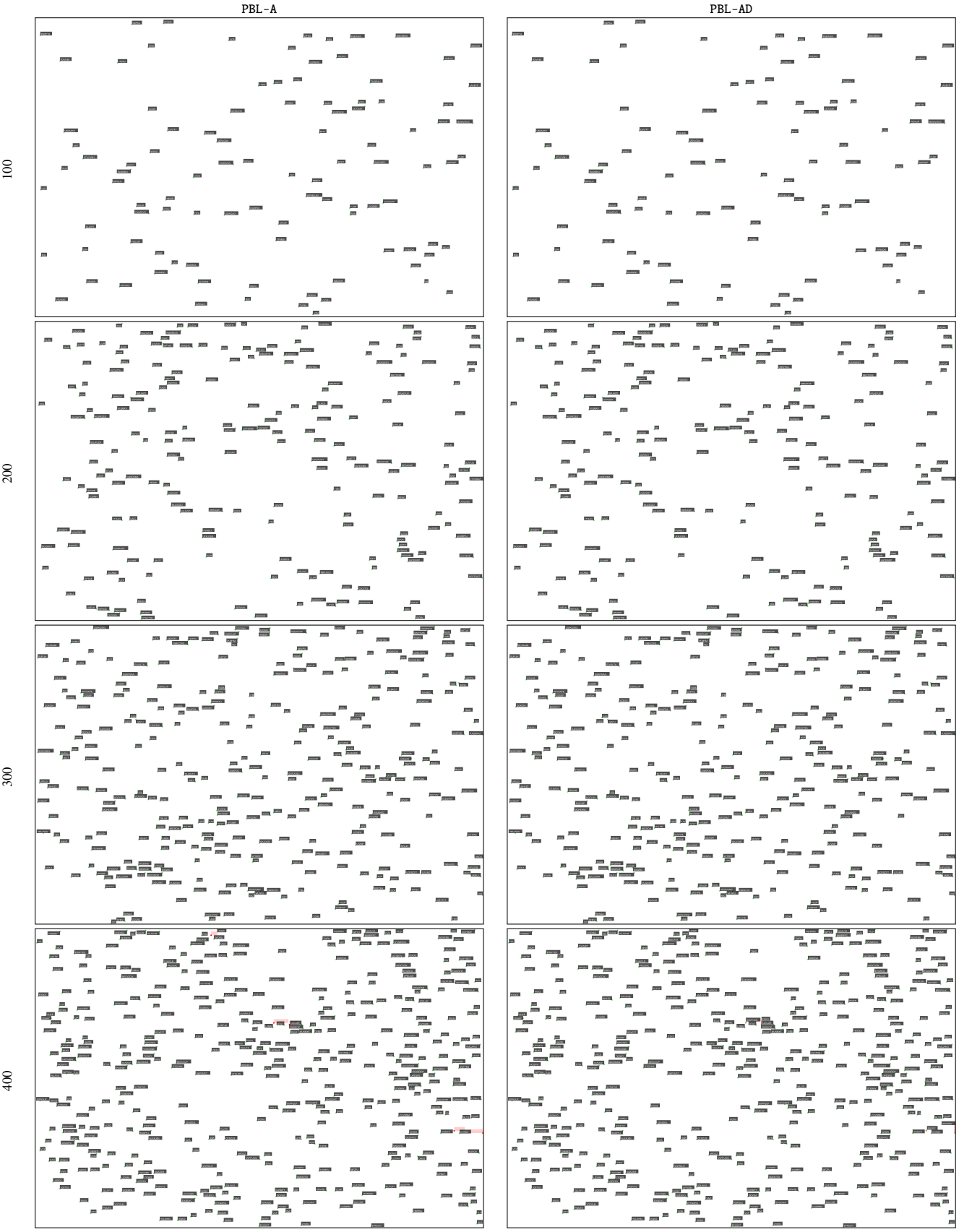
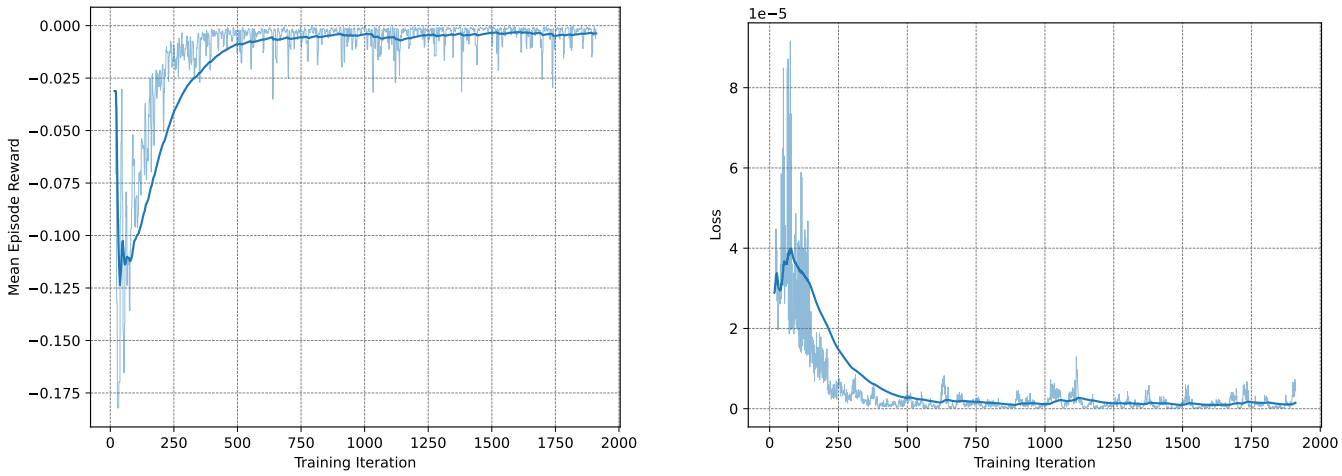


TABLE 7
Visual comparison of PBL-A and PBL-AD methods applied to selected instances from the volume dataset.



(a) (b)

Fig. 1. Training metrics of RFL (denoted by light blue) smoothed out using Exponential Moving Average (denoted by dark blue). Charts (a) and (b) show *mean episode reward* (same as mean return with discount factor $\gamma = 1.0$) and *loss* over within training iteration.