# A Experimental setup

## A.1 Synthetic data experiments

**Dataset.** As described in the main paper, the synthetic dataset (§3.4) is based on [34, §4.1]. The complexity of the dataset was increased by allowing two (instead of one) target training samples for each input. The probability $q(x)$ of sampling two targets instead of a single one for input $x$ is chosen as the piece-wise affine function such that $q(0) = q(1) = 1$ and $q(\frac{1}{2}) = 0$. This modification aimed to demonstrate how rMCL handles multiple targets sampled for each input during training.

**Architectures.** We use a three-layer perceptron backbone with 20 output hypotheses and a ReLu activation function for each layer and 256 hidden units. The multi-hypothesis splitting is carried out at the final layer stage. The scoring heads of the rMCL model receive the same representation as the hypothesis heads. The ensemble members have the same backbone but use a single hypothesis at the output stage. The rMCL$^*$ model has the same architecture as rMCL but utilizes a different training loss (see Section 3.2).

**Training details.** For each model, the training was performed on 100,000 training samples and 25,000 validation samples with a batch size of 1024. We employed 20 training epochs with Adam optimizer [24]. The checkpoint retained corresponded to the one with the lowest validation loss. For the *Stochastic Multiple Choice Learning* (sMCL) model, the multi-target version of the Winner-takes-all loss was used. The rMCL and rMCL$^*$ models were trained with unit scoring loss weight ($\beta = 1$, see §3.2), but the rMCL$^*$ training differed in that it only updated one negative hypothesis (compared to all in standard rMCL). When trained in this manner, the rMCL$^*$ model can be considered a more memory-efficient version of the proposed rMCL. The Independent Ensemble (IE) members were trained with a single target update (in the sMCL, the best hypothesis is updated for each target) as it resulted in a better fit to the data. The predictions from the IE members, trained with several initialization instances, were then stacked. The IE results were not plotted in Figure 1 for clarity and comparison purposes, as they were significantly worse (average Earth Mover's Distance at test time for the IE: $0.62 \pm 0.11$) compared to the other evaluated models.

**Evaluation details.** Figure 1, §3.4 (left) displays the Earth Mover's Distance (EMD) values using $\ell_2$ underlying distance calculated for 50 equally spaced input $x$ values in the comparative evaluation. At each input and for each model, the EMD was computed between $1,000$ samples taken from the ground-truth distribution, viewed as a mixture of Dirac deltas and the predicted hypothesis. The centroids in each cell (described in the right part of the figure) were computed using $35,000$ samples from the ground-truth distribution for each input.

## A.2 Audio data experiments

**Datasets preprocessing.** As indicated in §4.2, the experimental setup incorporates the ANSYN and RESYN datasets, which feature spatially localized events under anechoic and reverberant conditions respectively [1]. We used the first-order Ambisonics (FOA) format with four input audio channels. The events from 11 possible classes (Clearing throat, Coughing, Door knock, Door slam, Drawer, Human laughter, Keyboard, Keys put on a table, Page turning, Phone ringing and Speech), extracted from the DCASE16 Sound Event Detection dataset, were randomly placed in a spatial grid (see [1]). We adhered to the dataset preprocessing described by Schymura et al. in [37, 36]. The audio signals, with a sampling frequency of $44.1$ kHz, were converted into 30-second files. From those files, non-overlapping chunks of $0.5$ s were generated to be used as training inputs. Spectrogram computation was performed offline for saving computation, using Hann window with length $0.04$ s used for Short Term Fourier Transform (STFT) estimation, with 50% overlapping frames and 2048 Fast Fourier Transform (FFT) bins. The input for the model comprised both amplitude and phase information, stacked channel-wise.

**Architecture.** We employed the SELDNet backbone [1]. After raw audio preprocessing, it accepts the spectrograms of fixed duration with phase information as input and returns localization output in the chosen output resolution (here, $T = 25$ output time steps were considered for each chunk). The processing includes several feature extraction modules (CNNs, Bi-directional GRUs layers) that generate a representation at each time step in the output resolution. These latent representations are then mapped to the output localization estimates through FC layers. To accommodate the MCL setup, the final FC layers were split into $K$ FC heads, each producing a 2D output at each time step.

The rMCL architecture also includes full confidence heads at the output stage, each producing a scalar output in $[0, 1]$ through a sigmoid function. Therefore, the number of added parameters in the architecture is negligible ($\sim 4$ k when using $5$ output hypotheses for an architecture with $\sim 1.5$ M parameters).

**Training details.** The training setup is the same as the one used in [37, 36]. Unless specified otherwise, the training details correspond to those in [37], for which the official code was released, in particular, the implementation of the PIT variant used. The trainings were conducted using the AdamW optimizer [30], with a batch size of $128$, an initial learning rate of $0.05$, and following the scheduling scheme from [43]. The MCL models were trained using the multi-target version of the Winner-takes-all loss. The rMCL version was trained using confidence weight $\beta = 1$. As for the synthetic data experiments, the training of IE members was performed using different random seeds and single hypothesis loss with a single target update for each.

**Evaluation details.** As outlined in Section 4.2, the Oracle and EMD metrics with spherical distance were employed for evaluation purposes. The EMD metric is an extension to solve the assignment problem, often tackled using the Hungarian method [25]. Similar to prior studies (*e.g.*, [1, 37, 36]), the localization metric was computed solely for frames (the *active* frames) where at least one active source is present in the sound scene. For each test sample, the metrics were computed and averaged over the active frames among the output $T$ frames. Furthermore, we computed the standard deviation from the average metric for each subsection of the test set (D1, D2, and D3) sample-wise, as presented in the Tables. The standard deviations of the metrics when performing the exact same experiments from different random states are also provided in Appendix B.3. On a separate note, the frame recall metric, which indicates the percentage of time frames in which the number of active sound sources was estimated correctly, is omitted in the results for the sake of conciseness. This is because the EMD already penalizes missing sources in the predictions. In the rMCL model, the number of sources in the sound scene can nevertheless be computed before the normalization described in §3.2 by summing the output scores $\sum_k \gamma_\theta^k(\boldsymbol{x})$.

Visualizations of rMCL outputs for input test samples from the ANSYN dataset are given in Figure A.1.

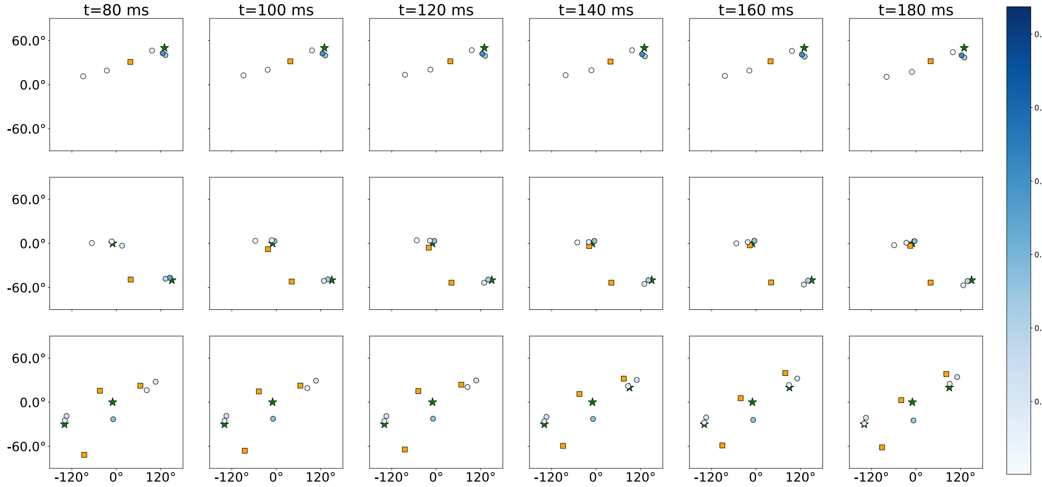


Figure A.1: **Qualitative comparisons**. Results for randomly chosen input audio clips (*different rows*) from the D3 test subset of the ANSYN dataset. The columns correspond to the temporal predictions at different time steps $t$. For each prediction (subplot), the abscissa and ordinates stand for azimuth and elevation angles, respectively (in degrees). We notice the competitive performance of the proposed rMCL model (with *shaded blue circles* for whose the score intensity is displayed in the colorbar) compared with the Permutation Invariant Training (PIT) approach (*orange squares*, baseline used for Tables 1 to 4) for predicting the positions of the targets (*green stars*).

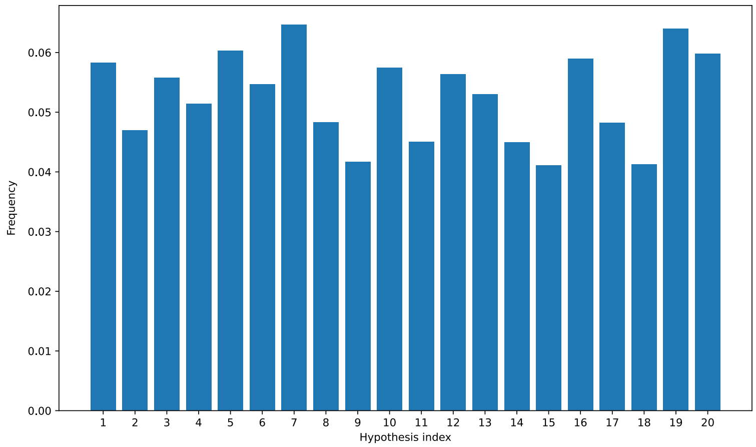


Figure B.1: **Histogram of *winner* hypotheses' indexes at test time for a 20-hypothesis model trained on ANSYN**. The x-axis displays the hypothesis index, and the y-axis the fraction of test samples on which hypothesis $k$ was selected. We observe a high entropy distribution, which shows that the collapse did not occur for this model.

### A.3 Computation details

We utilized the Hydra library for experimental purposes within the Pytorch deep learning framework. Our coding was inspired by [32, 37, 36, 1].

Our experiments were conducted on NVIDIA A100 GPUs. The total computing resources used for this project, including failed experiments, amount to approximately 2,000 GPU hours.

## B Further discussions

### B.1 About the collapse problem

As highlighted in the main paper, the *collapse* issue in MCL refers to a theoretical situation where one (or a few) of the $K$ hypotheses become dominant, *i.e.*, are almost always selected as a winner and receive the gradient update. In this situation, the other hypothesis heads are not updated, therefore shrinking the diversity of the predictions. As a way to measure the collapse phenomenon for a trained model $f_\theta = (f_\theta^1, \dots, f_\theta^K)$ using a validation dataset $\mathscr{D} \triangleq \{(\boldsymbol{x}_n, \boldsymbol{Y}_n)\}$, one can compute for each $k$, the number of samples in $\mathscr{D}$ for which hypothesis $k$ is a *winner*, that is:

$$\mathscr{N}_k(\mathscr{D}) \triangleq |\{(\boldsymbol{x}_n, \boldsymbol{Y}_n) \in \mathscr{D} : \exists \boldsymbol{y} \in \boldsymbol{Y}_n,\ \boldsymbol{y} \in \mathcal{Y}^k(\boldsymbol{x}_n)\}|.$$

The negative entropy of the histogram values $\{\mathscr{N}_k(\mathscr{D})\}_k$ should therefore inform about the collapse level; if the histogram shows a wide diversity of the selected hypotheses (*i.e.*, a histogram with almost uniform values for each bin), then there is no collapse. On the opposite, if the histogram has only one non-zero bin, the collapse level is maximum.

In our experiments with audio data, we did not observe the collapse problem in practice, neither with WTA nor with the proposed rMCL model. To verify this, one can compute the histogram of the $\{\mathscr{N}_k(\mathscr{D})\}$ values as explained above. See Fig. B.1 for an example of visualization for a 20-hypotheses model trained on ANSYN (corresponding to the results of Table 4 in the paper). As mentioned in [19] (p.8), we believe this issue is, in practice, solved by the variability of the data samples and the training stochasticity.

### B.2 Robustness in the presence of target outliers

This section aims to provide insights into how the proposed model can handle the presence of outliers in the output space, which may be critical in real-world datasets. Let's consider a setting where we have outliers in the training dataset, for instance, the toy use-case presented in the paper, where for

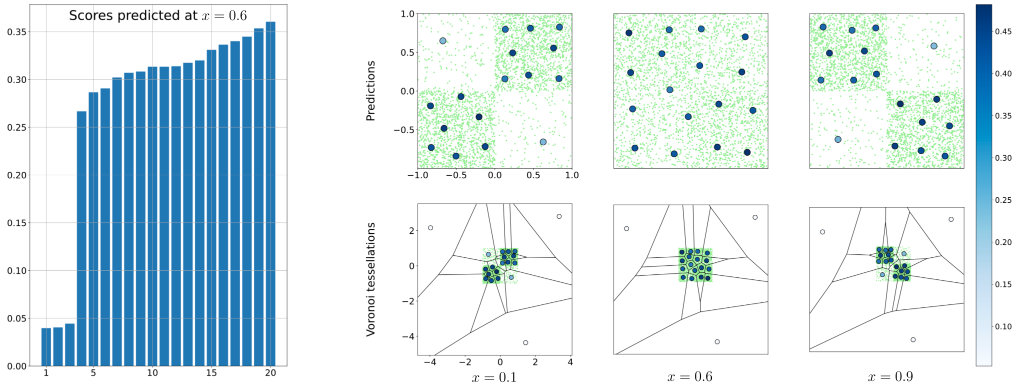


Figure B.2: **Robustness of rMCL to outliers**. Evaluation results on the toy example when corrupting the training dataset with outliers, modeled by a bivariate Cauchy distribution. (*Left*) Sorted values of the unnormalized scores predicted from the hypotheses scoring heads at $x = 0.6$. (*Right*) Inference at $x = 0.1, 0.6, 0.9$ with zoomed prediction in $[-1, 1]^2$ (*top*) and Voronoi tessellations in the full plane (*bottom*) with the same organization as in Figure 1.

each training example sampled, the probability of getting an outlier is $\rho \ll 1$. Then, whenever an outlier is sampled, one hypothesis will be pushed towards it with its associated score heads updated. As the training goes on, some of the hypotheses will manage the outlier samples; let's name them the 'outlier hypotheses'. Thanks to the proposed hypothesis scoring heads, the model will also learn the probability that an outlier hypothesis is chosen for a given training sample. Provided that the outlier likelihood is $\rho \ll 1$, the scoring heads will therefore prevent outlier hypotheses output from deteriorating the quality of the predicted distribution by rMCL. In Fig. B.2 an illustration of this phenomenon is proposed using a Cauchy distribution (we used $\rho = 0.02$, and corrupted samples having norm $\leq 2$ were rejected). We notice the above-explained phenomenon, where the so-called outlier hypotheses account for the outlier samples, while the other hypotheses lie in the square $[-1, 1]^2$ representing the samples from the ground-truth distribution.

Provided that the probability of sampling an outlier $\rho$ is small enough and the outliers are far enough from the ground-truth distribution to predict, the proposed rMCL model is therefore potentially robust to outliers. In this case, some specific hypotheses, namely the outlier hypotheses, will be assigned to them, preventing the non-outlier hypotheses from being heavily affected. At inference time, it will indeed be possible to set to zero the very low-score hypotheses given an arbitrary threshold so that the outlier hypotheses are not taken into account.

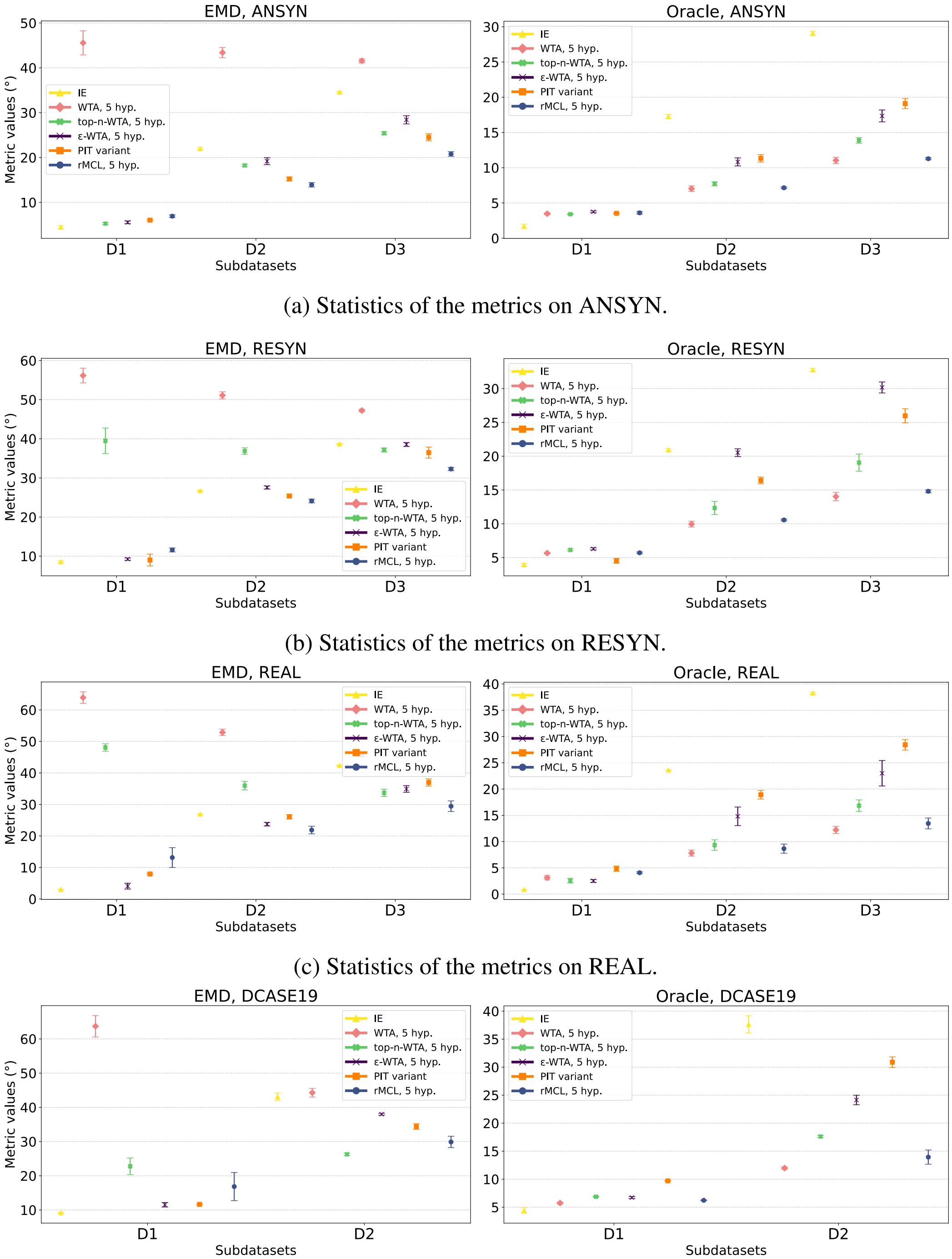
### B.3 Further results on audio datasets

We provide in this section further benchmarks of the method on sound source localization (SSL) datasets. Additionally to the results presented in Tables 1-4, where the mean and sample-wise standard deviation of the metrics on ANSYN and RESYN datasets are computed, we provide in Figures B.3 and B.4 further results considering the statistics of the metrics after several runs from different random states, and also including the REAL [1] and DCASE19 [3] datasets. In those datasets, the maximum number of overlapping events is respectively three and two.

**Further datasets.** In contrast to ANSYN and RESYN datasets which employ simulated Room Impulse Responses (RIRs) for audio spatialization, the REAL and DCASE19 datasets utilize RIRs recorded in real sound scenes. These were captured with a Spherical Microphone Array [1, 3]. Specifically, the recordings took place in various indoor settings inside a university. The REAL and DCASE19 RIRs were convoluted respectively with sound events from the UrbanSound8k [35] and DCASE16 datasets to achieve spatialization. Additionally, ambient noise was collected from the environment of the RIR recordings in the DCASE19 dataset. The same dataset pre-processing and sub-splitting process as presented in ANSYN and RESYN datasets was employed in REAL. For DCASE19, the four development dataset splits were consolidated to form the training set, while the official evaluation dataset served as the test set. The DCASE19 dataset comprises 60-second recordings. Notably, it has a low percentage of frames with overlapping events. To avoid bias from the dominance of monophonic events in some polyphonic recordings, the unimodal and multimodal splitting for the results in Figures B.3 and B.4 was executed at the prediction level, not the recording level, for this dataset. For each dataset, the WTA variants were trained with $n = 3$ and $\varepsilon = 0.5$.

The outcomes from both the REAL and DCASE19 datasets mirror the patterns identified in the previous datasets in Section 4.3. Within these results, the vanilla WTA approach, represented by the pink diamonds, marginally outperforms our proposed method on the oracle metric when an equivalent number of hypotheses is used. However, a substantial disparity is observed concerning the Earth Mover's Distance (EMD) metric when multiple hypotheses are predicted. Whenever a single source is present in the scene, the IE with five members (yellow line, triangles) still tend to outperform the other methods both in term of EMD and Oracle. In every multimodal setting within those datasets, we discern competitive results while comparing the EMD of rMCL (blue line, circles) and the other baselines.

Consistently with Section 4.5's analysis, increasing the number of hypotheses $K$ improves the oracle, but it may also degrade slightly the EMD when $K$ is too large. Furthermore, top-$n$-WTA shows a disparity of results across datasets. For instance, the EMD results of top-$n$ between ANSYN and other datasets reveal a contrary trend as the number of sources grows. Finally, it is noteworthy that, while oracle results remain stable, greater variability is seen in the rMCL unimodal EMD results in REAL and DCASE19 compared to prior datasets. This fluctuation may be attributed to the stochastic optimization sensibility to the challenging audio conditions, particularly when a single source is active and multiple hypotheses are utilized. Investigation on the explicit evaluation of the uncertainty estimated by rMCL, *e.g.*, due to possible label noise, is left for further work.



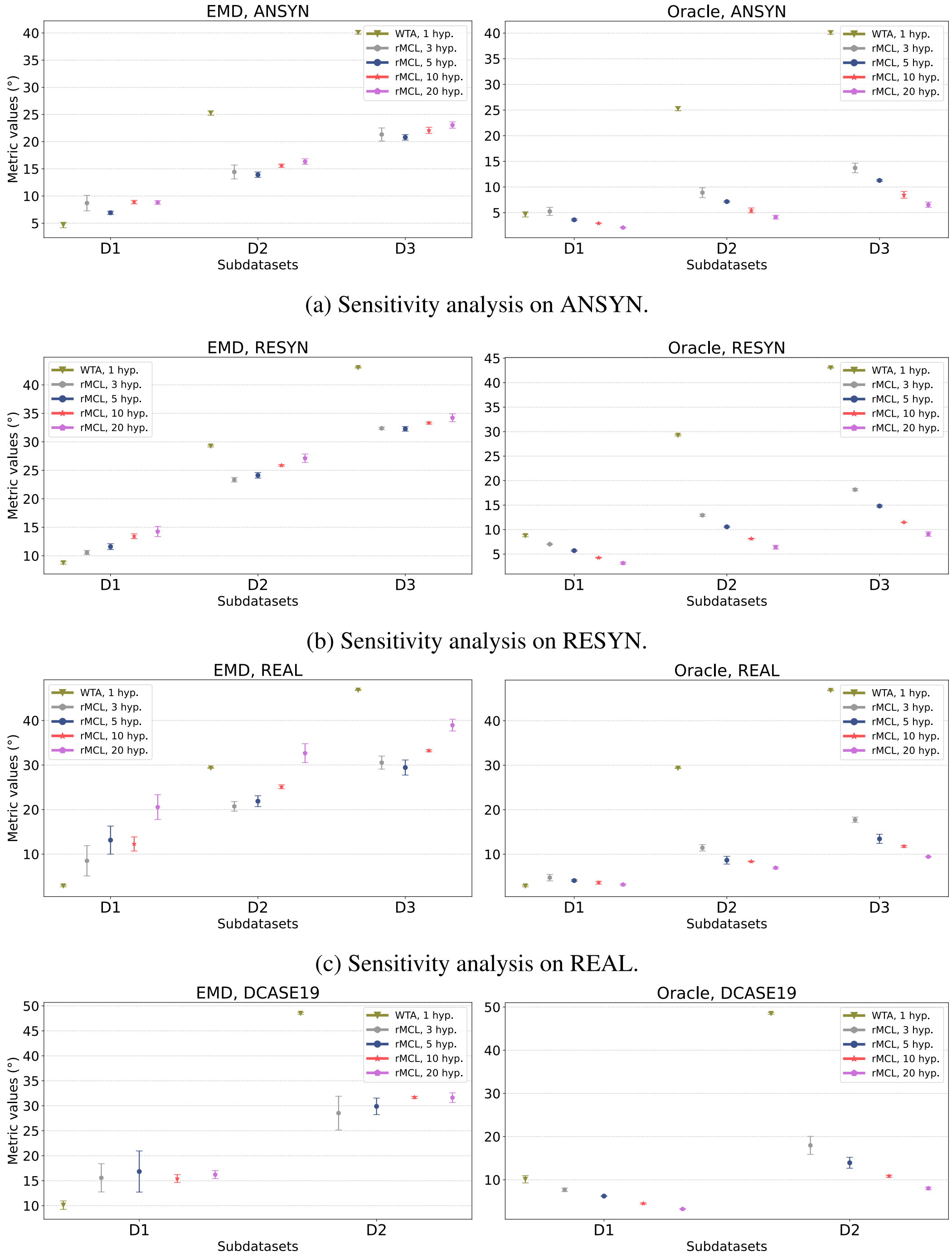
(a) Statistics of the metrics on ANSYN.

(b) Statistics of the metrics on RESYN.

(c) Statistics of the metrics on REAL.

(d) Statistics of the metrics on DCASE19.

Figure B.3: **Statistics of the metrics over four datasets**. Mean and standard deviation of EMD (*left*) and Oracle (*right*) over three training runs, on ANSYN, RESYN, REAL and DCASE19 datasets (*a-d*). Details and interpretation of the results are discussed in Sec. B.3.



(a) Sensitivity analysis on ANSYN.

(b) Sensitivity analysis on RESYN.

(c) Sensitivity analysis on REAL.

(d) Sensitivity analysis on DCASE19.

Figure B.4: **Effect of the number of hypotheses on the performance of rMCL**. Mean and standard deviation of EMD (*left*) and Oracle (*right*) over three training runs, on ANSYN, RESYN, REAL and DCASE19 datasets (*a-d*). Details and interpretation of the results are discussed in Sec. B.3.