

# impl TopKMeasure for ZeroConcentratedDivergence

Michael Shoemate

December 11, 2025

## 1 Hoare Triple

### Precondition

#### Compiler-verified

- Associated Const REPLACEMENT = true
- Method `privacy_map` Types consistent with pseudocode.

#### Caller-verified

- Method `privacy_map`
  - `d_in` is non-null and positive.
  - `scale` is non-null and positive.

### Pseudocode

```
1 # ZeroConcentratedDivergence
2 REPLACEMENT = True
3
4 def privacy_map(d_in: f64, scale: f64) -> f64:
5     return d_in.inf_div(scale).inf_powi(ibig(2)).inf_div(8.0)
```

### Postcondition

**Theorem 1.1.** The implementation is consistent with all associated items in the `TopKMeasure` trait.

1. Method `privacy_map`: For any  $x, x'$  where  $d_{in} \geq d_{Range}(x, x')$ , return  $d_{out} \geq D_{self}(f(x), f(x'))$ , where  $f(x) = \text{noisy\_top\_k}(x = x, k = 1, \text{scale} = \text{scale}, \text{replacement} = \text{Self} :: \text{REPLACEMENT})$ .

*Proof of postcondition: `privacy_map`.* [2] Proposition 2 shows that the exponential mechanism satisfies Bounded Range. [1] Lemma 3.2 shows that Bounded Range satisfies .  $\square$

## References

- [1] Mark Cesar and Ryan Rogers. Bounding, concentrating, and truncating: Unifying privacy loss composition for data analytics, 2020.
- [2] Jinshuo Dong, David Durfee, and Ryan Rogers. Optimal differential privacy composition for exponential mechanisms and the cost of adaptivity. *CoRR*, abs/1909.13830, 2019.