SAC-GNC: SAmple Consensus for adaptive Graduated Non-Convexity

Valter Piedade, Chitturi Sidhartha, José Gaspar, Venu Madhav Govindu, and Pedro Miraldo

MITSUBISHI ELECTRIC Changes for the Better — MITSUBISHI ELECTRIC RESEARCH LABORATORIES · TÉCNICO LISBOA · भारतीय विज्ञान संस्थान

ICCV HONOLULU HAWAII OCT 19-23, 2025

Project page

Graduated Non-Convexity (GNC)

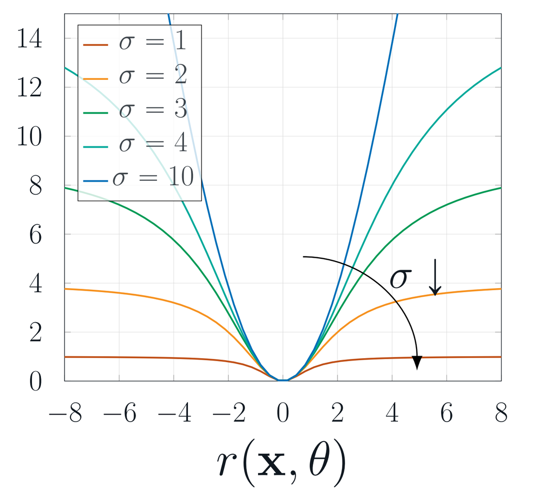
- ➤ GNC is a robust estimation algorithm that removes the dependency of the shape of the robust loss function by solving:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \rho_\sigma \left(r\left(\mathbf{x}_i, \theta\right)\right),$$

where $\rho_\sigma(\cdot)$ is, *e.g.*, the Geman-McClure loss;

- ➤ GNC is highly effective in computer vision tasks like 3D registration and pose graph optimization, where robustness is essential.

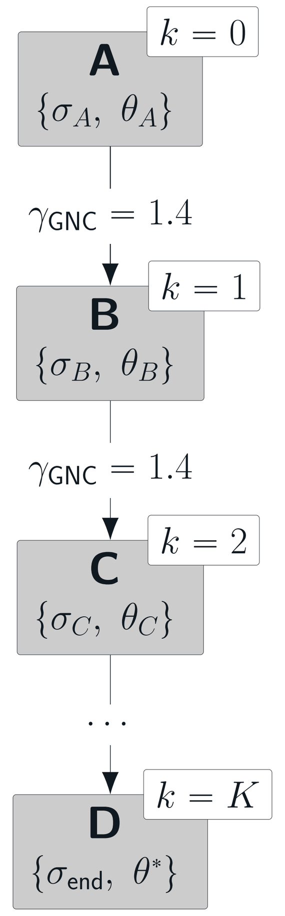
Geman-McClure loss



Principles of the GNC algorithm:

- At each iteration k, update σ_k according to $\sigma_k = \dfrac{\sigma_{k-1}}{\gamma_{\text{GNC}}}$, where γ_{GNC} is a fixed annealing factor;
- Iterate until a predefined σ_{end} is reached.

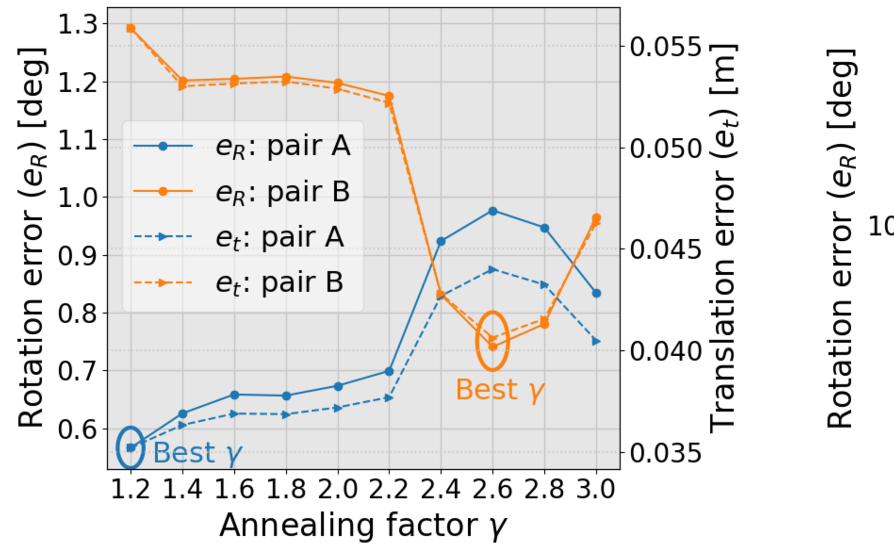
Diagram of GNC:



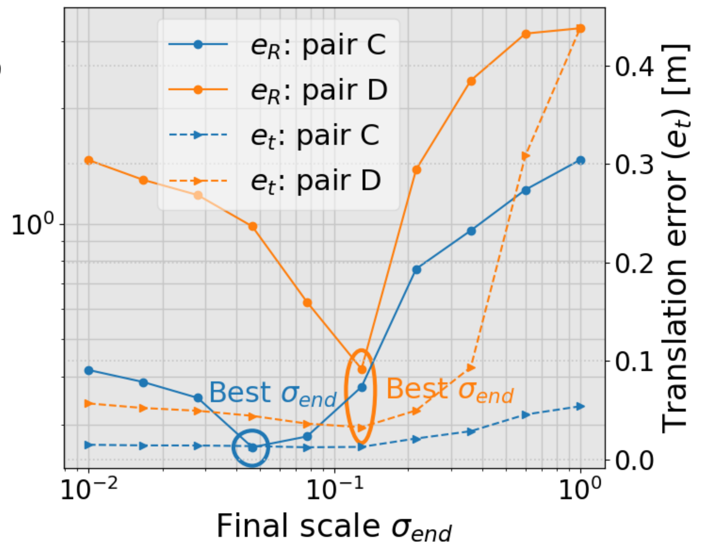
GNC's Limitations

- ➔ Previous GNC-based approaches use a fixed annealing factor γ_{GNC} and a predefined σ_{end};
- ➔ It is challenging to set γ_{GNC} and σ_{end} (data-dependent):

No "ideal" γ_{GNC}



No "ideal" σ_{end}



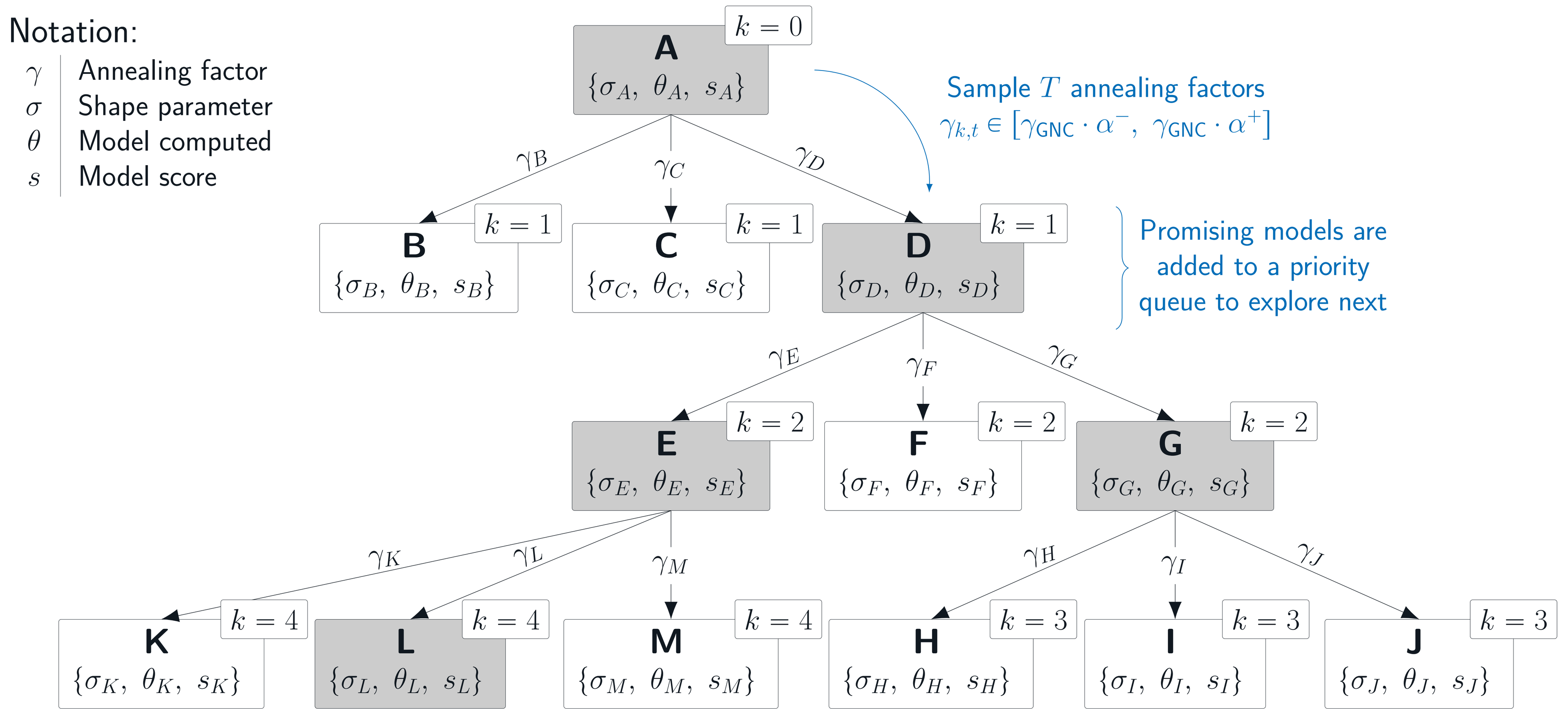
- ➔ Previous methods assume a continuous decrease of σ leads to the best solution, *i.e.*, the model estimate of the next iteration will be better than the current model.

Our Contributions

- ➤ A novel GNC-based adaptive annealing strategy for robust and efficient estimation;
- ➤ We show that combining sample & consensus (discrete) into GNC (continuous) has benefits over previous approaches;
- ➤ Experiments in real-world data in 3D registration and pose graph optimization show that SAC-GNC outperforms baselines in accuracy and efficiency.

SAmple Consensus for adaptive GNC

Diagram of the proposed *online searching strategy*:



Online search for σ

1. Annealing sampling – $\gamma_{k,t} \in [\gamma_{\text{GNC}} \cdot \alpha^-,\ \gamma_{\text{GNC}} \cdot \alpha^+]$, $t \in \{1, \cdots, T\}$;
2. Model scoring (*e.g.*, MSAC);
3. Priority queue search;
4. Save the best hypothesis.

- Relaxation ($\alpha^\pm$) of the fixed annealing factor γ_{GNC};
- Do not assume decreasing σ always leads to better solutions;
- Allows exploring various promising solutions.

Algorithm 1: *SAC-GNC*
Input – Data $\mathcal{D}$; annealing parameter γ_{GNC}; number of trial hypotheses T.
Output – Best hypothesis $\mathcal{H}^* = \{\sigma^*, \theta^*, s^*, d^*\}$.

```
k ← 0,  Q ← empty queue,  H_0 ← {σ_0, θ_0, ∞, 0};
θ_0 ← compute_initial_model (D);
σ_0 ← shape_initialization (θ_0);
while True do
    k ← k + 1;
    {σ_{k−1}, θ_{k−1}, s_{k−1}, d_{k−1}} ← H_{k−1};
    for t = 1 : T do
        γ_{k,t} ← get_annealing (γ_GNC, α^±);
        σ_{k,t} ← update_shape (σ_{k−1}, γ_{k,t});
        θ_{k,t} ← compute_model (D , θ_{k−1}, σ_{k,t});
        s_{k,t} ← compute_score (D, θ_{k,t});
        d_{k,t} ← d_{k−1} + 1;
        H_{k,t} ← {σ_{k,t}, θ_{k,t}, s_{k,t}, d_{k,t}};
    Q ← add_to_queue({H_{k,i}});
    H* ← save_best_hypothesis(H*, {H_{k,i}});
    if stopping_criteria(Q, H*) then
        break;
    H_k ← get_next_hypothesis(Q);
```

Stopping criteria

1. Check for model or scoring convergence;
2. Queue is empty.

- No need for a predefined σ_{end}.

Initialization

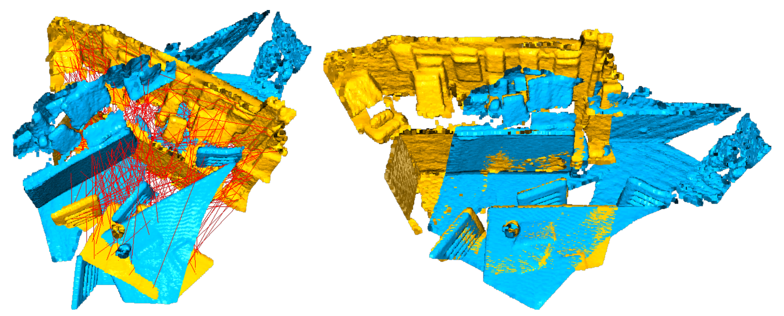
Compute a σ_0 that approximates a least-squares solution, by making the data point with maximum residual, r_{max}, weight ≈ 1:

$$w = \left(\frac{1}{1 + r_{\text{max}}^2/\sigma_0^2}\right)^2 \Leftrightarrow \sigma_0 = \frac{r_{\text{max}}}{\sqrt{(w^{-1/2} - 1)}}$$

Experiments and Results

➤ 3D registration

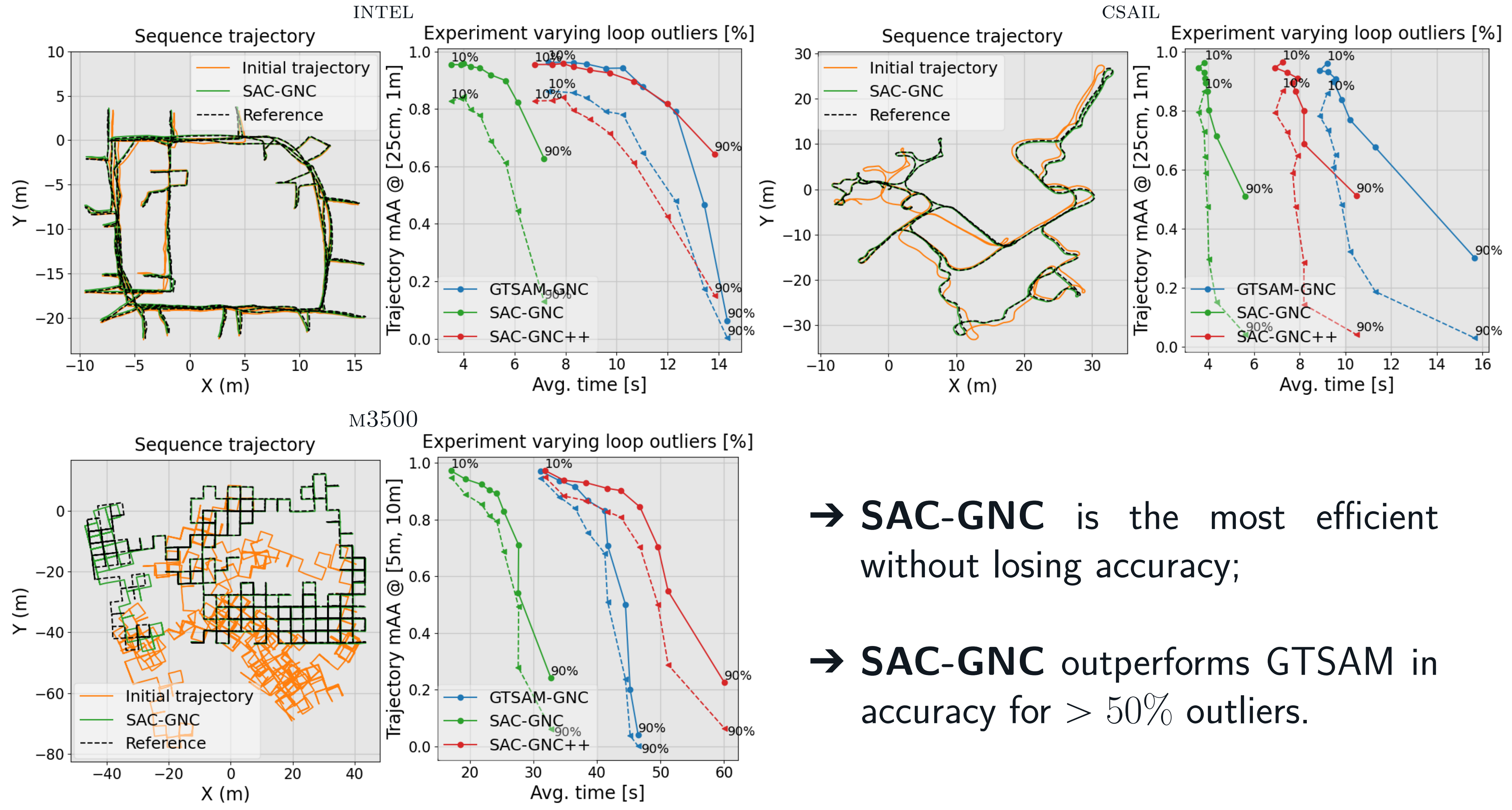
3DMatch dataset



- ➔ **SAC-GNC** handles outlier rates up to 90%.

Dataset	Method	mAA ↑ $(e_R, 5^\circ)$	mAA ↑ $(e_R, 10^\circ)$	mAA ↑ $(e_\mathbf{t}, 0.3m)$	mAA ↑ $(e_\mathbf{t}, 0.6m)$	Time [ms] ↓
↑ inliers ≈ 59.8%	RANSAC	0.492	0.707	0.713	0.829	5.13
	FGR	0.514	0.693	0.706	0.804	16.5
	TEASER++	Runtime fail (> 30 minutes per instance)				
	GNCp	0.553	0.726	0.738	0.831	5.61
	SAC-GNC	0.584	0.750	0.756	0.842	**3.98**
	SAC-GNC++	**0.586**	**0.753**	**0.759**	**0.846**	18.4
↓ inliers ≈ 11.6%	RANSAC	0.279	0.446	0.455	0.577	21.6
	FGR	0.271	0.417	0.443	0.549	12.6
	TEASER++	0.287	0.441	0.429	0.555	125
	GNCp	0.348	0.491	0.509	0.603	25.7
	SAC-GNC	0.421	0.539	0.544	0.616	**7.05**
	SAC-GNC++	**0.435**	**0.560**	**0.565**	**0.640**	56.6

➤ Pose graph optimization on SLAM sequences



- ➔ **SAC-GNC** is the most efficient without losing accuracy;
- ➔ **SAC-GNC** outperforms GTSAM in accuracy for $> 50\%$ outliers.

➤ Ablation studies

σ Search	Stopping Criteria	Initialization	mAA ↑ $(e_R, 5^\circ)$	mAA ↑ $(e_\mathbf{t}, 0.3m)$	Iter. ↓	Time [ms] ↓
			0.465	0.612	28	7.45
		✓	0.457	0.605	13.8	5.82
✓			0.487	0.619	11.9	7.47
✓	✓		0.487	0.618	10.9	6.80
✓		✓	0.490	0.622	7.83	6.69
✓	✓	✓	0.490	0.623	6.72	5.98

- ➔ σ search improves accuracy the most;
- ➔ Initialization and stopping criteria add efficiency without compromising accuracy.

Acknowledgments: Valter Piedade was supported by Mitsubishi Electric Research Laboratories, LARSyS funding (DOI: 10.54499/LA/P/0083/2020), and Sustainable Stone by Portugal (co-financed by the Recovery and Resilience Plan of the European Union). Pedro Miraldo was supported exclusively by Mitsubishi Electric Research Laboratories.