NEURAL INFORMATION PROCESSING SYSTEMS

Twelve Labs

Riiid

wrtn

EPFL

Addressing Negative Transfer in Diffusion Models

Hyojun Go*, Jinyoung Kim*, Yunsung Lee*, Seunghyun Lee*,
Shinhyeok Oh, Hyeongdon Moon, Seungtaek Choi†

▸ Twelve Labs ▸ Wrtn ▸ Riiid ▸ EPFL *Co-first author †Corresponding Author

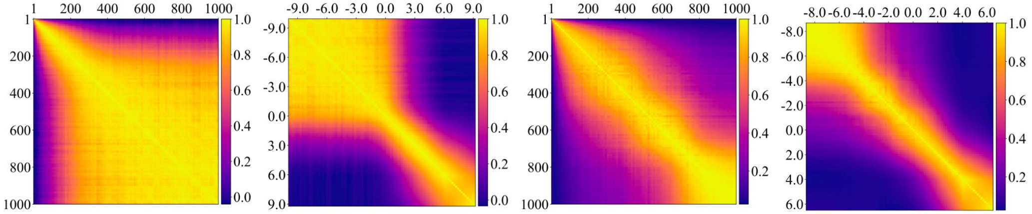
Rethinking Diffusion Model as Multi-Task Learning

- We first rethink the diffusion model as multi-task learning where each task corresponds to denoising tasks at different timestep

D^t : Denoising task at timestep t learned by $L_t = ||\epsilon - \epsilon_\theta(\mathbf{x}_t, t)||_2^2.$

Analyzing Diffusion Models from Multi-Task Learning

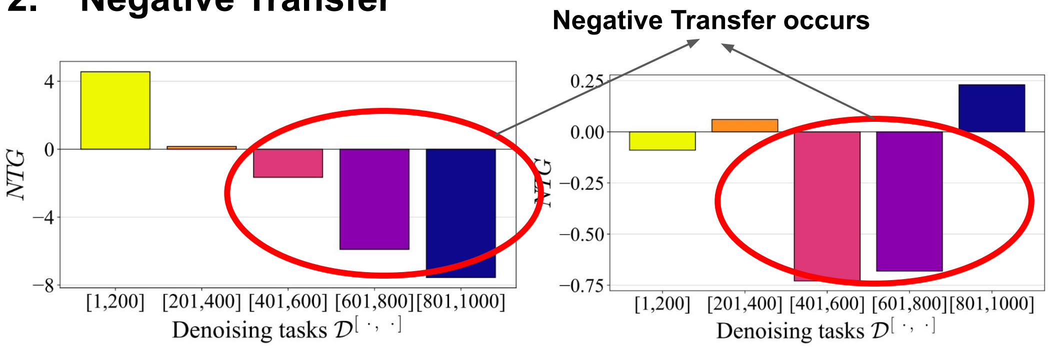
1. Task Affinity: Gradient direction-based task affinity score



(a) ADM (timestep t) (b) ADM (log-SNR) (c) LDM (timestep t) (d) LDM (log-SNR)

1) We observe that the task affinity for denoising tasks decreases as the discrepancy between noise level and timestep increases.
2) This suggests that tasks sharing temporal/noise-level proximity can be cooperatively learned without significant conflict.

2. Negative Transfer



(a) ADM (b) LDM

1) Negative transfer refers to deterioration in a multi-task learner performance due to conflicts between tasks.
2) It can be identified by observing the performance gap between a multi-task and specific-task learner.
3) We define NTG for this, when NTG < 0, negative transfer occurs, showing that a multi-task learner underperform than a specific task learner.

Leveraging MTL approach

To remediate negative transfer, we leverage well-established MTL methods.

1. **Gradient conflicts: PCgrad** [1] mitigate conflicting gradients between tasks by projecting conflicting parts of gradients.

2. **Gradient balancing: NashMTL** [2] balances gradients between tasks by solving a bargaining game.

3. **Loss weighting: Uncertainty Weighting (UW)** [3] balances task losses by weighting each task loss with task-dependent uncertainty.

[1] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. Advances in Neural Information Processing Systems, 33:5824–5836, 2020.
[2] Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pages 16428–16446.
[3] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7482–7491, 2018.

Interval Clustering for Grouping Denoising Tasks

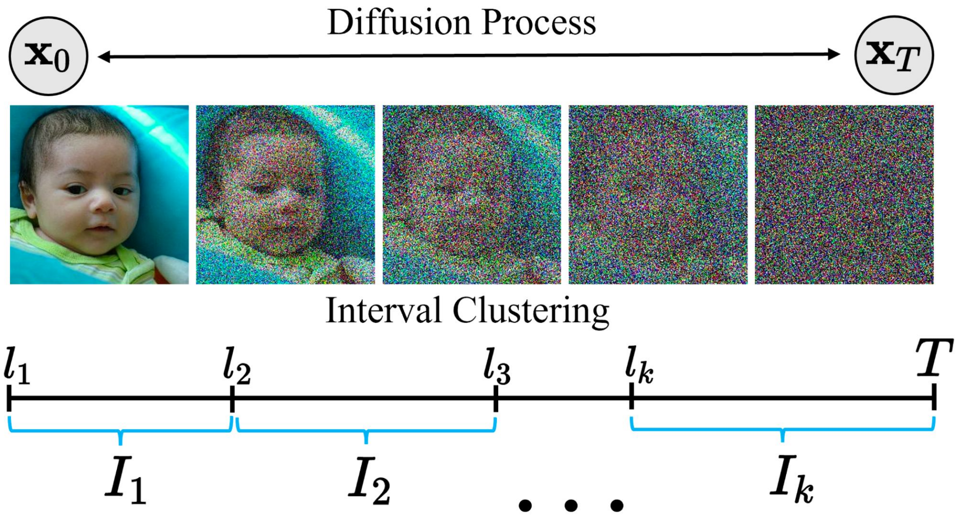
- MTL methods can require **a large amount of computation**, especially when the number of tasks is large.
- To address this, we leverage an interval clustering algorithm to **group denoising tasks** with interval clusters inspired from task affinity

$\mathcal{X} = \{1, \ldots, T\}$ to k contiguous intervals $I_1, \ldots, I_k$, where $I_i = [l_i, r_i]$ and $l_i \leq r_i$.

$$\min_{l_1=1<l_2<\ldots<l_k} \sum_{i=1}^{k} L_{cluster}(I_i \cap \mathcal{X})$$

For clustering object, we propose

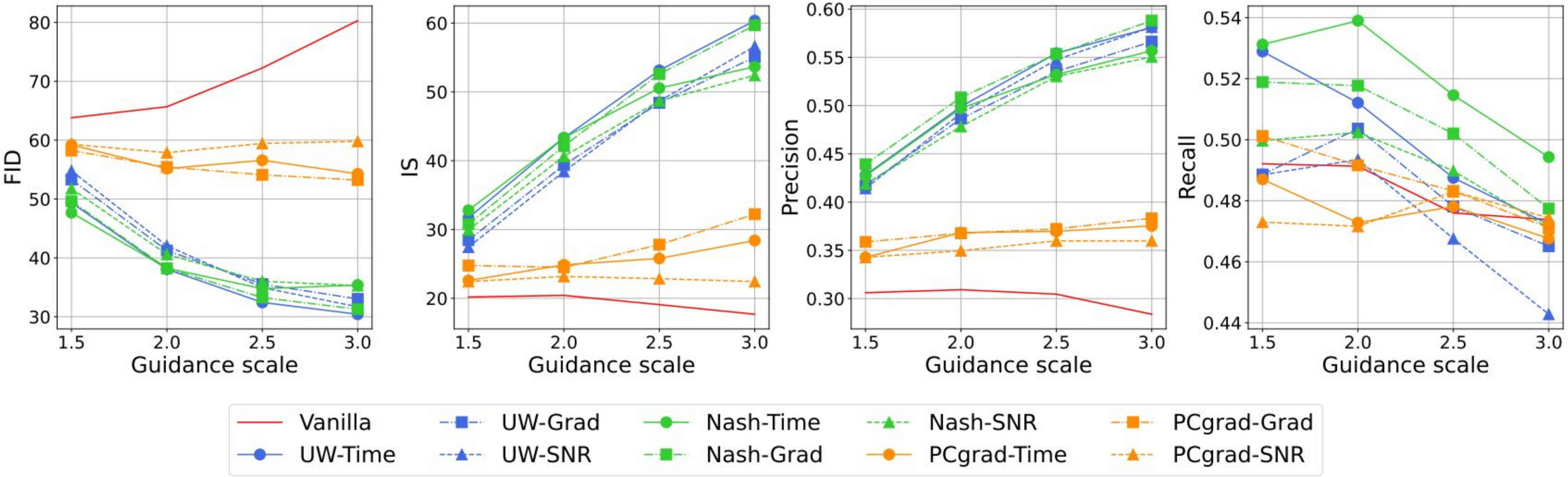
1. Timestep-based Clustering
2. SNR-based Clustering
3. Gradient-based Clustering



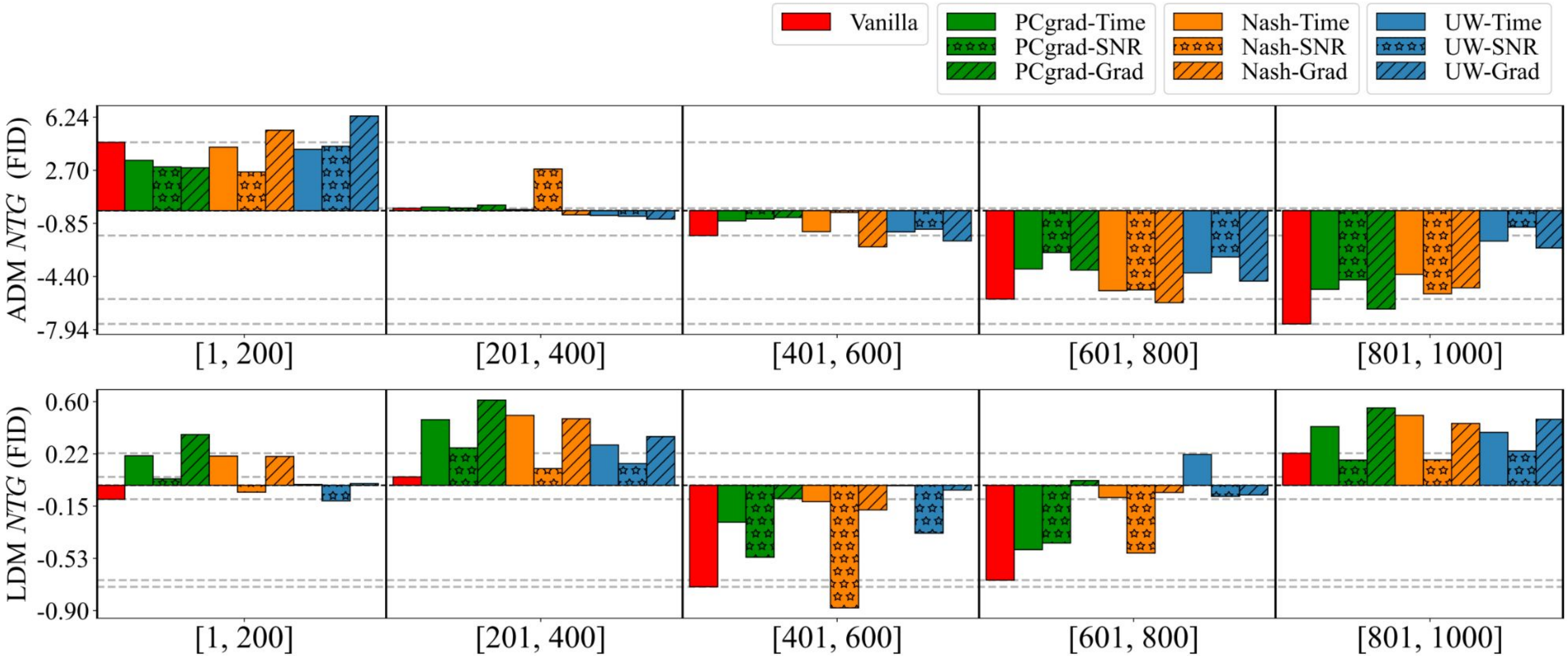
Results: Improvement of Diffusion Performance

Model	Clustering	Method	Dataset					
			FFHQ [26]			CelebA-HQ [24]		
			FID (↓)	Precision (↑)	Recall (↑)	FID (↓)	Precision (↑)	Recall (↑)
ADM [7, 6]		Vanilla	24.95	0.5427	0.3996	22.27	0.5651	0.4328
	Timestep	PCgrad [75]	22.29	0.5566	0.4027	21.31	0.5610	0.4238
		NashMTL [41]	21.45	0.5510	**0.4193**	20.58	0.5724	0.4303
		UW [27]	20.78	0.5995	0.3881	**17.74**	**0.6323**	0.4023
	SNR	PCgrad [75]	20.60	0.5743	0.4026	20.47	0.5608	0.4298
		NashMTL [41]	23.09	0.5581	0.3971	20.11	0.5733	**0.4388**
		UW [27]	**20.19**	**0.6297**	0.3635	18.54	0.6060	0.4092
	Gradient	PCgrad [75]	23.07	0.5526	0.3962	20.43	0.5777	0.4348
		NashMTL [41]	22.36	0.5507	0.4126	21.18	0.5682	0.4369
		UW [27]	21.38	0.5961	0.3685	18.23	0.6011	0.4130
LDM [50]		Vanila	10.56	0.7198	0.4766	10.61	0.7049	0.4732
	Timestep	PCgrad [75]	9.599	0.7349	0.4845	9.817	0.7076	0.4951
		NashMTL [41]	9.400	0.7296	0.4877	9.247	0.7119	0.4945
		UW [27]	9.386	0.7489	0.4811	9.220	0.7181	0.4939
	SNR	PCgrad [75]	9.715	0.7262	0.4889	9.498	0.7071	0.5024
		NashMTL [41]	10.33	0.7242	0.4710	9.429	0.7062	0.4883
		UW [27]	9.734	0.7494	0.4797	**9.030**	**0.7202**	0.4938
	Gradient	PCgrad [75]	**9.189**	0.7359	0.4904	10.31	0.6954	0.4927
		NashMTL [41]	9.294	0.7234	**0.4962**	9.740	0.7051	**0.5067**
		UW [27]	9.439	**0.7499**	0.4855	9.414	0.7199	0.4952

Results: Comparison in Class-Conditional Generation



Results: Reduced Negative Transfer Gap



Highlighted Results: ANT-UW

Our method, ANT-UW, that employ UW with interval clustering greatly outperforms MinSNR. 2. ANT-UW needs similar computation and memory cost to Vanilla training.

Table 2: Comparison between MinSNR and ANT-UW. DiT-L/2 is trained on ImageNet.

Method	FID	IS	Precision	Recall
Vanilla	12.59	134.60	0.73	0.49
MinSNR	9.58	179.98	0.78	**0.47**
ANT-UW	**6.17**	**203.45**	**0.82**	**0.47**

Table 3: GPU memory usage and runtime comparison on FFHQ dataset in LDM architecture.

Method	GPU memory usage (GB)	# Iterations / Sec
Vanilla	34.126	**2.108**
PCgrad	**28.160**	1.523
NashMTL	38.914	2.011
UW	34.350	2.103

Project page:
https://gohyojun15.github.io/ANT_diffusion/

Code:
https://github.com/gohyojun15/ANT_diffusion