DMTG: One-Shot Differentiable Multi-Task Grouping

Yuan Gao, Shuguo Jiang, Moran Li, Jin-Gang Yu, Gui-Song Xia

Our Code is Released!
https://github.com/ethanygao/DMTG

Motivation

Address Multi-Task Learning (MTL) with a large number of tasks by Multi-Task Grouping (MTG).

Given N tasks, we identify the best task groups from 2^N candidates and train the model weights simultaneously in one-shot, with the high-order task affinity fully exploited.

Key Ideas & Features

We formulate MTG as a fully **differentiable pruning** problem on an **adaptive network architecture** determined by an unknown categorical distribution. Our method exhibits the following features:

- Group identification is formulated as **learning a relaxed categorical distribution** rather than heuristics.
- **One-shot training** eliminates the objective bias between group identification and model grouped task learning.
- **pruning formulation** instead of sampling group candidates to train from scratch

Accuracy

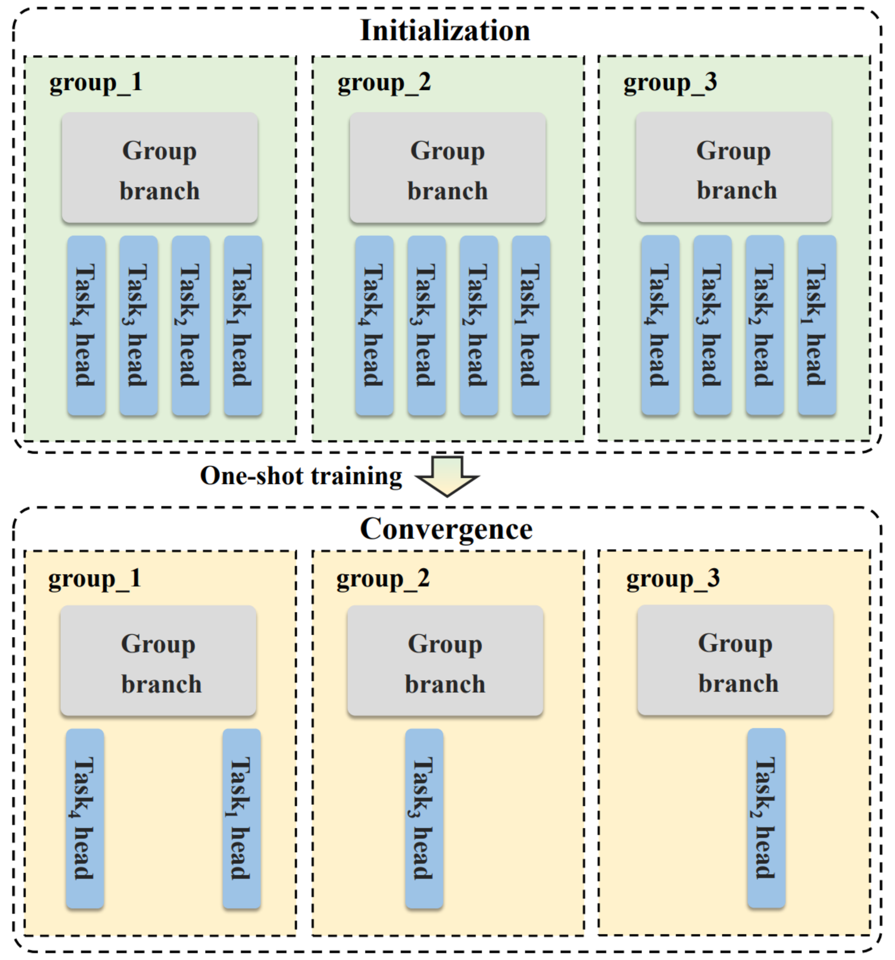
Efficiency

MTG as Network Pruning

Initialization: Each group connects to all the task heads, ensuring full exploration of high order task-affinity.

Convergence: Learn a categorical distribution to *exclusively and uniquely* categorize each task into one group.

One-shot Training: *Simultaneously* prune task heads and train the weights of group-specific branches.



Formulation & Architecture & Optimization

Grouping Formulation as Categorical Distribution:

➤ Each task is *exclusively and uniquely* belongs to one group.

For example, categorize N Tasks $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}$ into K Groups $\{\mathcal{G}_1, \ldots, \mathcal{G}_K\}$

N Tasks $\mathcal{T} = \cup_{k=1}^{K} \mathcal{G}_k$, K Groups

$$s.t. \quad \forall k, \quad |\mathcal{G}_k| \in \{0, \ldots, N\}, \quad \forall (i,j), \quad \mathcal{G}_i \cap \mathcal{G}_j = \emptyset,$$

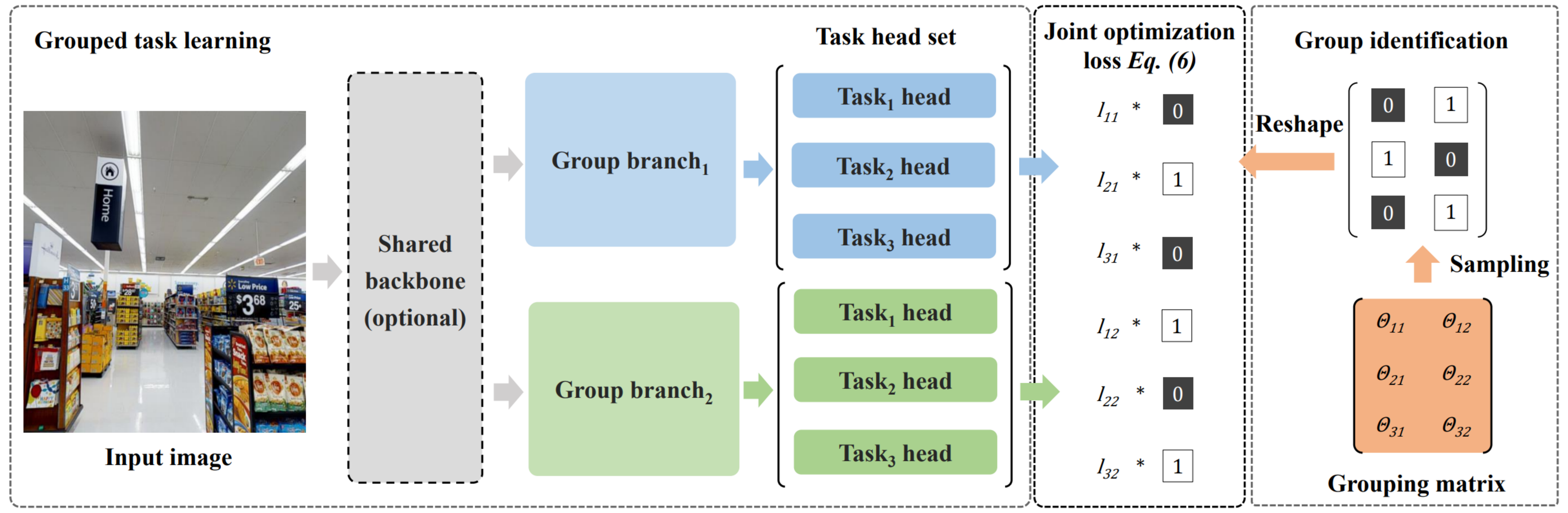
➤ Learn a set of random variables $Z = \{z_{ik}\} \in \mathbb{R}^{N \times K}$, where z_{ik} categorize task i to group k.

$$\sum_k z_{ik} = 1, \quad \text{and} \quad z_{ik} \in \{0, 1\}, \quad \forall k,$$

$$z_{ik} \sim \texttt{Categorical}(s_{ik})$$

➤ Continuous Relaxation by *Gumbel Softmax*.

$$\tilde{z}_{ik} = \frac{\exp((s_{ik} + g_{ik})/\tau)}{\sum_{m=1}^{K} \exp((s_{im} + g_{im})/\tau)}$$



Architecture & Optimization (for categorizing *N* tasks into *K* groups):

➤ **Grouped task learning module.** We use K branches in the grouped task learning module, each linked to N task heads, optimizing high-order task affinity with efficient *O(K)* training complexity, and further reduce complexity with optional group-wise shared layers.

➤ **Group identification module.** The categorization of N tasks into K groups involves learning an unknown categorical distribution, which determines an adaptive network architecture and can be optimized jointly with model weights in a one-shot pruning problem.

➤ **Joint optimization.** The discrete categorical distribution is continuously relaxed, allowing joint optimization of group identification parameters and grouped task learning weights in one-shot using gradients from the task loss. The continuous relaxation is facilitated by the reparameterization trick from the concrete distribution and the *Gumbel Softmax*.

Experiments

Compared with SOTAs:

Taskonomy dataset with 5 tasks (Taskonomy-9)

Groups	Methods	Total Loss ↓	NormGain$_L$ (%) ↑	Relative Train. Complex.
-	Naive MTL	0.223	-	1
-	STL	0.199	+57.35	$O(N)$
$K=3$	RG	0.231	-21.73	$O(K)$
	HOA	0.190	+47.78	$O(N^2)+O(K)$
	TAG	0.210	+45.02	$O(N)+O(K)$
	MTG-Net	0.191	+57.83	–
	Ours	**0.173**	**+63.85**	$O(K)$
$K=4$	RG	0.204	+19.90	$O(K)$
	HOA	0.195	+47.95	$O(N^2)+O(K)$
	TAG	0.190	+49.41	$O(N)+O(K)$
	MTG-Net	0.191	+57.83	–
	Ours	**0.170**	**+64.58**	$O(K)$
$K=5$	RG	0.198	+28.40	$O(K)$
	HOA	0.195	+47.95	$O(N^2)+O(K)$
	TAG	0.190	+49.41	$O(N)+O(K)$
	MTG-Net	0.191	+57.83	–
	Ours	**0.168**	**+65.01**	$O(K)$

CelebA dataset with 9 tasks (CelebA-9)

Groups	Methods	Total Error ↓	NormGain$_E$ (%) ↑	Relative Encoder Complex.
-	Naive MTL	56.13	-	1
-	STL	59.93	-8.70	$O(N)$
$K=2$	RG	54.87	+1.06	$O(K)$
	HOA	53.60	+3.27	$O(N^2)+O(K)$
	TAG	53.41	+4.38	$O(N)+O(K)$
	Ours	**52.97**	**+5.75**	$O(K)$
$K=3$	RG	54.57	+1.54	$O(K)$
	HOA	54.04	+3.62	$O(N^2)+O(K)$
	TAG	54.37	+2.08	$O(N)+O(K)$
	Ours	**53.67**	**+4.64**	$O(K)$
$K=4$	RG	54.57	+1.54	$O(K)$
	HOA	54.14	+2.53	$O(N^2)+O(K)$
	TAG	54.11	+3.17	$O(N)+O(K)$
	Ours	**53.62**	**+4.62**	$O(K)$

Taskonomy dataset with 5 tasks (Taskonomy-5) *w.r.t* each input task

Groups	Methods	Depth Estimation		Surface Normal		Semantic Segmentation		Keypoint Detection		Edge Detection	
		Loss ↓	NormGain$_L$ (%) ↑	Loss ↓	NormGain$_L$ (%) ↑	Loss ↓	NormGain$_L$ (%) ↑	Loss ↓	NormGain$_L$ (%) ↑	Loss ↓	NormGain$_L$ (%) ↑
-	Naive MTL	8.67e-3	-	1.07e-1	-	8.28e-2	-	1.19e-2	-	1.31e-2	-
-	STL	1.60e-5	+99.82	1.07e-1	-0.18	9.16e-2	-10.63	1.30e-4	+98.91	1.56e-4	+98.81
$K=3$	RG	2.57e-2	-195.88	1.08e-1	-0.81	8.43e-2	-1.88	6.87e-3	+42.46	6.88e-3	+47.45
	HOA	5.85e-3	+32.47	1.11e-1	-4.37	7.33e-2	+11.49	**2.00e-6**	**+99.98**	8.60e-5	+99.34
	TAG	5.15e-3	+40.59	1.21e-1	-12.93	8.43e-2	-1.88	**2.00e-6**	**+99.98**	8.60e-5	+99.34
	MTG-Net	2.04e-4	+97.65	**1.07e-1**	**+0.00**	8.28e-2	+0.00	6.39e-4	+94.65	4.08e-4	+96.88
	Ours	**1.19e-7**	**+100.00**	1.07e-1	-0.05	**6.65e-2**	**+19.64**	4.30e-5	+99.63	**3.58e-7**	**+100.00**
$K=4$	RG	5.15e-3	+40.59	1.07e-1	+0.00	7.33e-2	+11.49	1.19e-2	+0.00	6.88e-3	+47.45
	HOA	5.15e-3	+40.59	1.06e-1	+0.44	8.33e-2	-0.61	**2.00e-6**	**+99.98**	8.60e-5	+99.34
	TAG	5.15e-3	+40.59	1.11e-1	-4.37	7.33e-2	+11.49	**2.00e-6**	**+99.98**	8.60e-5	+99.34
	MTG-Net	2.04e-4	+97.65	1.07e-1	+0.00	8.28e-2	+0.00	6.39e-4	+94.65	4.08e-4	+96.88
	Ours	**1.19e-7**	**+100.00**	**1.05e-1**	**+1.43**	**6.46e-2**	**+21.96**	4.70e-5	+99.61	**1.20e-5**	**+99.91**
$K=5$	RG	5.15e-3	+40.59	1.07e-1	+0.00	7.33e-2	+11.49	6.87e-3	+42.46	6.88e-3	+47.45
	HOA	5.15e-3	+40.59	1.06e-1	+0.44	8.33e-2	-0.61	2.00e-6	+99.98	8.60e-5	+99.34
	TAG	5.15e-3	+40.59	1.11e-1	-4.37	7.33e-2	+11.49	2.00e-6	+99.98	8.60e-5	+99.34
	MTG-Net	2.04e-4	+97.65	1.07e-1	+0.00	8.28e-2	+0.00	6.39e-4	+94.65	4.08e-4	+96.88
	Ours	**1.19e-7**	**+100.00**	**1.05e-1**	**+1.62**	**6.34e-2**	**+23.43**	**1.00e-6**	**+99.99**	**4.17e-7**	**+100.00**

Ablation Analysis:

- Our proposed MTG outperforms two-shot methods given the *same group categorization*.

Groups	Methods	Total Loss ↓	NormGain$_L$ (%) ↑
$K=3$	Retrain from Scratch	0.183	+53.34
	Retrain from Naive MTL Init.	0.194	+57.10
	Ours (one-shot)	**0.173**	**+63.85**
$K=4$	Retrain from Scratch	0.183	+53.34
	Retrain from Naive MTL Init.	0.194	+57.10
	Ours (one-shot)	**0.170**	**+64.58**
$K=5$	Retrain from Scratch	0.190	+59.47
	Retrain from Naive MTL Init.	0.194	+58.87
	Ours (one-shot)	**0.168**	**+65.01**

- Our method generalizes to different backbones including *CNNs* and *transformers*.

Backbone	Methods	Total Loss ↓	NormGain$_L$ (%)
ViT-Base	Naive MTL	0.453	-
	STL	0.435	+58.72
	HOA	0.379	+62.22
	TAG	0.439	+58.72
	MTG-Net	0.403	+47.65
	Ours	**0.326**	**+68.31**

- Our method scales to a large number of input tasks, i.e., CelebA dataset with ***40*** tasks.