LEARNING HOW TO SHARE: OPTIMAL TRANSPORT FOR MEDICAL MULTI-TASK LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We present ASK-OT (**AS**ymmetric **K**nowledge sharing via **O**ptimal **T**ransport), an innovative framework for multi-task learning that formulates knowledge sharing as a resource allocation problem. ASK-OT learns a set of knowledge modules from a shared representation and allocates them to tasks according to their representational needs. We formulate this allocation as an optimal transport problem, allowing task-specific and asymmetric sharing patterns to emerge from the learned representations rather than being imposed by predefined task groups. To account for the distinct roles of modules and tasks, we introduce a module-capacity constraint and task constraint relaxation, which respectively allow modules to be shared across tasks while allowing tasks to draw different amounts of knowledge. We theoretically characterize the resulting asymmetric sharing and demonstrate experimentally that ASK-OT outperforms representative baselines. Code will be available upon acceptance.

1 INTRODUCTION

Multi-task learning (MTL) is a key strategy for building efficient models that handle multiple tasks within a single framework (Caruana, 1997). This is especially valuable in medical imaging, where annotated data is costly to obtain and multiple clinically related findings can often be extracted from the same imaging study (Zhao et al., 2023; Bragman et al., 2019). By training related tasks jointly, MTL enables the model to learn shared knowledge representations and capture the latent relationships between tasks (Evgeniou & Pontil, 2004; Thung & Wee, 2018). It also reduces computational overhead compared to training individual models for each task. However, since each task has its own objective function, improving one task's performance can come at the expense of others, a phenomenon known as negative transfer.

To mitigate negative transfer, existing approaches primarily operate at the level of optimization, by reweighting task losses or modifying conflicting gradients (Liu et al., 2023; Navon et al., 2022; Liu et al., 2021a; Ban & Ji, 2024; Liu et al., 2021b). Yet conflicting task objectives can also degrade the shared latent representation, indicating that task conflict is not only an optimization problem but can also affect the learned representation (Huang et al., 2026). A separate line of work focuses on the sharing structure by grouping tasks according to estimated relatedness (Jeong & Yoon, 2025). However, grouping imposes discrete sharing relationships and cannot account for asymmetric sharing between tasks. Task relatedness does not necessarily imply equal benefit from shared representations, as knowledge useful for one task may provide only limited benefit to another. Consequently, related tasks may require asymmetric knowledge sharing. This motivates learning task-specific allocations of shared representations, rather than imposing fixed and symmetric sharing relationships.

Motivated by this, we formulate knowledge sharing in MTL as a knowledge resource allocation problem on the latent space: a finite set of non-linear *knowledge modules* provides representational sources, while tasks have different demands for those sources. Built on top of a shared encoder, these modules provide distinct views of the shared representation, while each task is equipped with a learnable embedding that captures its representational needs. Intuitively, each module acts as a bounded lens on the shared representation that can recombine what the encoder produces to emphasize the needs of the tasks. We formulate the allocation of modules to tasks as an optimal transport (OT) problem (Peyré & Cuturi, 2019), where the resulting transport plan determines how much each

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

task draws from each module based on their similarity. This formulation allows the sharing structure to be learned from tasks' needs, rather than fixed in advance (See Figure 2(a)).

Standard OT formulations, however, require both sides of the transport plan to match, which is well suited to comparing two distributions of the same kind but not to our setting, where knowledge modules and tasks play fundamentally different roles. On the modules side, a single, most-compatible task may absorb most of a module's capacity, leaving other tasks without access to its knowledge. We therefore impose a *Module-Capacity Constraint (MCC)* by scaling each module's capacity to the number of tasks. As a module's scaled capacity exceeds what any single task can absorb, MCC forces the excess onto other tasks according to compatibility. Thus, sharing emerges as a structural outcome of the optimization.

On the tasks side, a hard constraint that requires each task to receive a fixed total allocation from the knowledge modules can be restrictive, as it may force a task to draw knowledge even when the available modules do not sufficiently match its needs, particularly when well-matched modules have limited capacity. We therefore introduce *Task Constraint Relaxation (TCR)*, allowing each task to draw more or less knowledge from modules according to its compatibility with the available modules. Together, MCC and TCR allow the sharing structure to be both distributed and task-dependent: MCC prevents a module from being fully absorbed by a single task, while TCR allows each task to draw knowledge according to how useful the available modules are to the task. Consequently, tasks can rely on overlapping modules to different degrees, allowing knowledge sharing to emerge asymmetrically between tasks. Unlike grouping-based sharing, which imposes discrete and symmetric relationships, our formulation supports flexible and directional knowledge sharing. We formalize this property theoretically in Section 5. Our contributions are threefold.

- We present ASK-OT (**AS**ymmetric **K**nowledge sharing via **O**ptimal **T**ransport), a formulation of representation-level knowledge sharing in MTL as an optimal transport-based allocation problem. Learnable knowledge modules serve as representational sources, while tasks draw from these sources according to their representational needs, allowing the sharing structure to emerge rather than being fixed a priori.
- We introduce MCC and TCR constraints that account for the distinct roles of modules and tasks in the allocation problem. MCC promotes sharing of each module across multiple tasks, while TCR allows tasks to rely on the available modules to different degrees, enabling flexible and asymmetric knowledge sharing.
- We provide theoretical insights into the asymmetric knowledge sharing of ASK-OT, and evaluate it experimentally across medical and general-domain benchmarks, showing that it outperforms representative baselines across multiple MTL paradigms.

2 Related Work

Multitask learning. Prior works in MTL span several directions. *Loss-balancing* methods reweight task losses: FAMO (Liu et al., 2023) approximates a Pareto-descent direction without full gradient access, and Nash-MTL (Navon et al., 2022) frames the update as a bargaining game among tasks. *Gradient-conflict mitigation* methods instead modify the joint update direction: CAGrad (Liu et al., 2021a) constrains conflicts among task gradients, FairGrad (Ban & Ji, 2024) incorporates fairness in balancing task improvements, and IMTL (Liu et al., 2021b) jointly accounts for loss scale and gradient direction. *Task grouping* methods group tasks according to their relationships: selTG (Jeong & Yoon, 2025) dynamically partitions tasks based on inter-task affinity and sequentially updates the resulting groups. In contrast to loss-balancing and gradient-conflict methods, which reweight losses or gradients while keeping the shared representation fixed across tasks, we act directly on the representation. Each task uses a task-specific recombination of shared latent features, with sharing patterns emerging continuously from its needs rather than being imposed by discrete group assignments, as in task grouping.

Architecture-based approaches instead control how tasks access shared parameters. Mixture-of-experts models such as MMoE (Ma et al., 2018) provide a bank of shared experts and learn task-specific gates, while Hazimeh et al. (2021) introduces sparse and differentiable expert selection. Suteu & Serban (2026) similarly retains a largely shared feature extractor while introducing task-specific gates through normalization. In these methods, each task's gate is learned independently,

with no mechanism linking its allocation to what other tasks draw from the same pool. Ours instead jointly optimizes all task–module allocations under a single transport plan, so sharing patterns emerge from the competing demands of tasks rather than being decided one task at a time.

Optimal transport in machine learning. OT provides a principled way to construct correspondences between distributions while accounting for the cost of moving mass (Villani & et al., 2008; Peyré & Cuturi, 2019). OT has been applied to representation alignment tasks such as cross domain alignment (Li et al., 2024a), noisy-label learning (Feng et al., 2023), federated learning (Li et al., 2024b), and vision-language modeling, aligning either the full set of representations (Chen et al., 2023) or a partial subset (Shiri et al., 2025). We instead use OT for *knowledge resource allocation* between modules and tasks, where the two constraints play structurally distinct roles rather than a symmetric alignment role, producing the asymmetric sharing patterns central to our formulation.

3 PRELIMINARIES

In MTL, tasks are often grouped based on relatedness and then share representations jointly. This implies that related tasks that are grouped share and benefit mutually. In this section, we first assess this assumption empirically on a set of clinically related tasks, then introduce optimal transport.

Analyzing task relationships and transfer patterns. Before introducing ASK-OT, we examine whether clinical relatedness predicts how tasks benefit from shared representations. Fluorescein angiography (FA) provides a suitable test bed because multiple clinically distinct findings are assessed from the same imaging, providing shared visual information while retaining different clinical objectives (Sheridan et al., 2010). We therefore compare single-task learning (STL) with all multi-task combinations of three FA tasks: choroidal neovascularization (CNV), hyperfluorescence type (HyperF), and hypofluorescence type (HypoF) (full details in Appendix A).

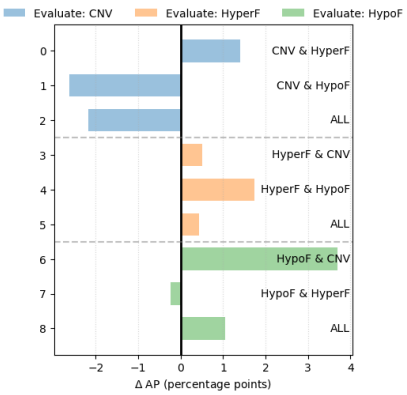


Figure 1: AP gain (percentage points) over STL, averaged over five seeds. Bars are grouped by the evaluated task (color), each label on the right lists the tasks trained jointly. Positive values indicate that joint training improves the evaluated task.

The results are summarized in Figure 1, which reports the change in average precision (AP) relative to STL for each evaluated task under different sharing configurations. Each bar corresponds to one evaluated task, with the jointly trained task(s) indicated next to it. For example, the top bar measures how much CNV improves when trained together with HyperF rather than alone, whereas the fourth bar measures the effect of the same pairing on HyperF. The results reveal two patterns. First, semantic relatedness is not a reliable predictor of transfer: CNV and HyperF are clinically related and transfer positively for both, but CNV and HypoF, which are not closely related, produce an even larger gain for HypoF. Second, transfer is also not symmetric: when mutual, its magnitude is unequal (CNV benefits more than HyperF from their pairing), and it can even be one-sided, e.g. pairing CNV–HypoF helping HypoF while harming CNV. Consistent with this, grouping all tasks together is never the best configuration for any task. Moreover, each task's optimal partner differs: CNV pairs best with HyperF, HyperF with HypoF, and HypoF with CNV, so no fixed grouping can satisfy all three tasks at once. Together, these results show that a priori task knowledge may not be sufficient to decide what to share, and that discrete group assignments are inherently limited, motivating mechanisms that learn continuous, task-specific, asymmetric sharing patterns.

Optimal Transport. OT finds the efficient way to transport mass between two distributions. In the discrete setting, let $\mathbf{u} \in \mathbb{R}^M$ and $\mathbf{v} \in \mathbb{R}^N$ be the source and target distributions, and $C \in \mathbb{R}^{M \times N}$ be the cost matrix where $c_{i,j}$ determines the cost of transferring mass from u_i to v_j. OT seeks a transport plan Π that minimizes the total cost. As directly solving the OT is time-consuming, Sinkhorn algorithm (Cuturi, 2013) adds an entropic regularization for fast optimization:

$$\Pi^* = \arg\min_{\Pi \in U(\mathbf{u},\mathbf{v})} \langle C, \Pi \rangle_F - \epsilon H(\Pi), \tag{1}$$

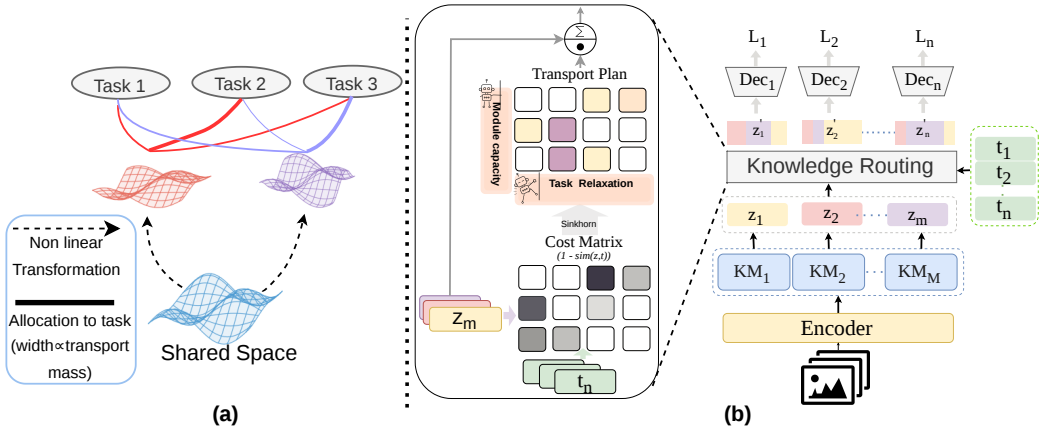


Figure 2: **(a) Conceptual illustration:** Non-linear knowledge modules transform the shared space into distinct representations, from which each task draws a different amount based on the OT plan (arc width $\propto$ transport mass $\Pi^\star$), yielding asymmetric sharing. **(b) Overall pipeline:** Task representations $\{z'_i|_{i=1}^N\}$ are formed as a $\Pi^\star$-weighted combination of knowledge module outputs $\{z_i|_{i=1}^M\}$. $\Pi^\star$ is computed via an optimal transport problem (subject to MCC and TCR) over module–task compatibility, and then passed to task-specific decoders.

where $\langle\cdot,\cdot\rangle_F$ denotes the Frobenius inner product, $H(\cdot)$ denotes the entropy of the transport plan, ϵ controls the strength of the regularization, and $U(\mathbf{u},\mathbf{v})$ is the set of feasible transport plans satisfying the marginal constraints:

$$(a) : \Pi \mathbf{1}_N = \mathbf{u}, \qquad (b) : \Pi^\top \mathbf{1}_M = \mathbf{v}, \quad \Pi \in \mathbb{R}_+^{M\times N} \tag{2}$$

both constraints are typically set as uniform discrete distributions ($u = \mathbf{1}_M/M$, $v = \mathbf{1}_N/N$) (Chen et al., 2023), imposing the same total mass. The resulting transport plan Π^* provides a soft assignment between elements of the two distributions.

4 Method

Overview. Multiple tasks are considered with their own loss function denoted as $\{\mathcal{L}_i|_{i=1}^N\}$, where N represents the number of tasks. The network parameters are partitioned into three groups: 1) shared encoder ($f(.)$), which is shared by all tasks, 2) learnable *knowledge modules*, $\mathcal{KM} = \{\mathcal{KM}_i|_{i=1}^M\}$ on top of f and 3) task-specific decoders ($\{Dec_i|_{i=1}^N\}$). MTL is formulated as a resource allocation problem, where $\mathcal{KM}$ serves as sources of knowledge, and tasks serve as targets. Each task uses a combination of the knowledge provided by all modules. This combination is calculated by a soft transport plan Π^*, where $\Pi^*_{m,n}$ denotes the amount of knowledge allocated from $\mathcal{KM}_m$ to task n.

Knowledge modules. Each knowledge module $\mathcal{KM}_m$ is a learnable transformation that operates on the shared encoder features. For an input image x, each module produces a transformed representation $\mathcal{KM}_m(f(x)) =: \mathbf{z}_m \in \mathbb{R}^d$. Specialization emerges through training as the transport plan Π^* determines how strongly each task's gradient influences each module. So modules naturally specialize toward the tasks that rely on them, while tasks that require conflicting transformations are steered toward different modules. This mitigates the gradient interference of naive parameter sharing while preserving positive transfer among tasks with compatible needs.

Task representations. To encode the task n's knowledge requirements in the shared representation space, a learnable embedding $\mathbf{t}_n \in \mathbb{R}^d$ is defined and learned jointly with Π and $\mathcal{KM}$. Since $\mathcal{T} = \{t_i|_{i=1}^N\}$ is a part of the end-to-end optimization, it evolves according to the supervision signal of its corresponding task. As a result, $\mathbf{t}_n$ act as a representative of its corresponding objective $\mathcal{L}_n$.

Module Capacity and Task Relaxation. To reflect the different roles that modules and tasks play in our setting, we modify the constraints in OT formulation (equation 2). Treating each $\mathcal{KM}$ as a finite shared resource, we impose a *Module-Capacity Constraint (MCC)* on the rows of the transport plan, scaling each module's capacity to the number of tasks:

$$\text{MCC:} \quad u_m = \frac{N}{M} \ \forall m, \qquad \Pi \mathbf{1}_N = \mathbf{u} \tag{3}$$

MCC prevents a module's mass from being fully absorbed by a single task, structurally driving its allocation across multiple tasks according to module–task compatibility.

Additionally, the column constraint of OT governs how each task draws knowledge across modules. Enforcing $\mathbf{v}$ as a hard equality as in equation 2(b), forces every task to receive the same total amount of knowledge, regardless of how well the available modules match its needs. We therefore relax this constraint by replacing hard equality with a soft penalty on the deviation of $\Pi^\top \mathbf{1}_M$ from $\mathbf{v}$:

$$\text{TCR:} \quad \Pi^* = \underset{\Pi \mathbf{1}_N = \mathbf{u}}{\arg\min} \langle C, \Pi \rangle_F - \epsilon H(\Pi) + \epsilon \frac{\tau}{1-\tau} \mathrm{KL}\big(\Pi^\top \mathbf{1}_M \,\|\, \mathbf{v}\big), \quad \tau \in [0, 1) \tag{4}$$

where KL stands for Kullback–Leibler, and τ controls the weight of penalty. As $\tau \to 1$, $\Pi^\top \mathbf{1}_M = \mathbf{v}$ is forced exactly and recovers the hard equality. As $\tau \to 0$, the penalty vanishes and each task's allocation is instead governed by C. We refer to this as *Task Constraint Relaxation (TCR)*.

Rather than being imposed externally, the task allocation now emerges from a trade-off between v_n and minimizing the transport cost $\langle C_{:,n}, \Pi_{:,n} \rangle_F$: if the cost term dominates, $\Pi_{m,n}$ is shaped primarily by task–module similarity, but if the penalty dominates, $\Pi_{m,n}$ is pulled back toward satisfying v_n. For instance, if task n is well-matched only to module i but $\mathcal{KM}_i$ lacks the capacity, the high cost of drawing from mismatched modules outweighs the KL penalty, so the plan tolerates $[\Pi^\top \mathbf{1}_M]_n \neq v_n$ rather than importing irrelevant knowledge. Whereas if n is well-matched to many modules, the KL penalty dominates, and $[\Pi^\top \mathbf{1}_M]_n \approx v_n$ is recovered.

Knowledge sharing via Optimal Transport. To assess the alignment between the knowledge produced by each module and the knowledge requirements of task n, we compute $sim(\mathbf{z}_m, \mathbf{t}_n)$, where $sim(\cdot, \cdot)$ denotes cosine similarity. The resulting similarities are converted into transport costs as $C_{m,n} = 1 - \mathrm{sim}(\mathbf{z}_m, \mathbf{t}_n)$. Given the cost matrix and the proposed transport constraints, the resulting OT problem is solved using a Sinkhorn-style algorithm (Cuturi, 2013), adapted to our asymmetric constraints following unbalanced OT (Chizat et al., 2018) (see Appendix B). The resulting transport plan $\Pi^\star$ determines the contribution of each knowledge module to each task and is used to form task-specific representations: $z'_n = \sum_{m=1}^{M} \Pi^\star_{m,n} z_m$, which are then used by the corresponding task decoders. The overall pipeline is shown in Figure 2(b).

5 Theoretical Insights

Formulating knowledge sharing as an OT problem gives two properties: (i) the transport plan is uniquely defined by a strictly convex optimization problem, and (ii) MCC ensures balanced module utilization (Details in Appendix D). Here, we show how *asymmetric* knowledge sharing emerges across tasks and modules: ***MCC*** structurally induces sharing, while ***TCR*** regulates its allocation through transport costs and task saturation. Finally, we show that sharing between two tasks is not constrained to have equal magnitudes in the two directions. Throughout, asymmetry refers to unequal sharing magnitudes.

Lemma 1 (Forced spillover)**.** *Consider the hard-constrained limit $\tau \to 1$, in which the transport plan satisfies $\Pi^* \mathbf{1}_N = \mathbf{u}, \quad (\Pi^*)^\top \mathbf{1}_M = \mathbf{v}$, with $u_m = N/M$ and $v_n = 1$. Let $PT(m)$ denote the preferred task of module m (i.e. the task with the lowest cost), and define*

$$r_{PT(m)} := v_{PT(m)} - \sum_{j \neq m} \Pi^*_{j,PT(m)}$$

If $r_{PT(m)} < u_m$, then module m must allocate a positive amount of mass to at least one non-preferred task: $\sum_{n \neq PT(m)} \Pi^*_{m,n} > 0$.

When $M < N$, this condition holds for every module since $r_{PT(m)} \leq 1 < N/M = u_m$. Thus, MCC structurally enforces knowledge sharing: each module must distribute its capacity across multiple tasks, while the transport costs determine where the spillover is allocated. The proof is given in Appendix C.1.

Lemma 2 (Relative sharing under soft TCR)**.** *Consider the soft TCR formulation with $\tau < 1$, and let $a_n = \sum_{m=1}^{M} \Pi^*_{m,n}$ denote the total mass allocated to task T_n. For a fixed module m and two destination tasks T_i and T_j, the relative amount of mass allocated by module m to these tasks*

satisfies

$$\frac{\Pi^*_{m,i}}{\Pi^*_{m,j}} = \exp\left(-\frac{C_{m,i} - C_{m,j}}{\epsilon}\right)\left(\frac{v_i/a_i}{v_j/a_j}\right)^{\frac{\lambda}{\epsilon}}, \qquad \lambda = \epsilon\frac{\tau}{1-\tau}$$

Consequently, $\Pi^*_{m,i} > \Pi^*_{m,j}$ *if and only if*

$$C_{m,i} - C_{m,j} < \lambda \log\left(\frac{v_i/a_i}{v_j/a_j}\right)$$

So, allocation is governed jointly by module–task cost and task saturation: lower cost favors a task, while lower relative saturation can strengthen this preference. See Appendix C.2 for the derivation.

Proposition 1 (Decoupled directional sharing)**.** *For two distinct tasks* T_i *and* T_j*, the directional sharing magnitudes*

$$\Phi(i \to j) = \sum_{m:PT(m)=i} \Pi^*_{m,j}, \qquad \Phi(j \to i) = \sum_{m:PT(m)=j} \Pi^*_{m,i}$$

are governed by different preferred-module sets, module–task costs, and task-saturation factors. The optimization imposes no constraint requiring $\Phi(i \to j) = \Phi(j \to i)$*.*

So, the two directional sharing magnitudes need not be equal. The derivation is in Appendix C.3.

6 Experiments

Experimental settings. We evaluate on two medical datasets with different relationships to the pretrained encoder. AMD (Fang et al., 2022) uses the same imaging modality as the pretraining data, but does not overlap with it, providing an *in-domain* setting. FA (Xu et al., 2025) covers the same organ and clinical domain but uses an unseen imaging modality, providing an *out-of-domain* setting. To assess performance beyond medical imaging and scalability with respect to the number of tasks, we additionally evaluate on CelebA (Liu et al., 2015) under two configurations: the 9-task subset introduced by Fifty et al. (2021) and the full 40-task benchmark. Task performance is measured using average precision (AP) for classification tasks and mean absolute error (MAE) for regression task. To compare across all tasks, the relative performance improvement over the standard joint learning baseline is used following Maninis et al. (2019): $\Delta_{method} = \frac{1}{N}\sum_{i=1}^{N}(-1)^{l_i}(M_{method,i} - M_{b,i})/M_{b,i}$, where $M_{b,i}$ is the baseline, and $l_i = 1$ if lower values indicate better performance and $l_i = 0$ otherwise. To capture whether gains occur jointly across metrics rather than on isolated tasks, we additionally report the Hypervolume (HV) indicator, the volume of the joint $[0,1]^n$ metric space dominated by a model's Pareto front relative to the worst-case corner (Özbulak et al., 2025). Details in Appendix E.

Architectures. We used a shared encoder with task-specific decoders. For the encoder, we adopted RETFound-Green (Engelmann & Bernabeu, 2025) on the medical datasets and DINOv2-Base (Oquab et al., 2024) on CelebA. Each task decoder is a single linear layer. $\mathcal{KM}$ is implemented as a self-attention module.

Baselines. We compare our method against representative MTL approaches spanning several paradigms: single-task learning (STL), standard joint training with gradient descent (GD), loss-balancing methods such as FAMO (Liu et al., 2023) and Nash-MTL (Navon et al., 2022), STCH (Lin et al., 2024), gradient conflict mitigation methods such as CAGrad (Liu et al., 2021a) and FairGrad (Ban & Ji, 2024), hybrid approaches such as IMTL (Liu et al., 2021b), task grouping methods such as selective tasks grouping (selTG, Jeong & Yoon (2025)) , and architecture-based parameter-sharing methods such as MMoE (Ma et al., 2018), and TSσLN (Suteu & Serban, 2026).

6.1 Comparison

Medical Domain Evaluation. Table 1 shows that our method provides the strongest overall performance across the two medical benchmarks. On FA, it achieves the highest Δ (+4.46%) and HV (0.040), outperforming the second best, Nash-MTL (+2.96%, 0.036) and FAMO (+2.49%, 0.037). On AMD, which contains imbalanced tasks, the advantage is more pronounced, with ASK-OT reaching +13.07% Δ and 0.162 HV, compared with +12.15% and 0.110 for CAGrad, the second

Table 1: Results on medical datasets (mean±std over 5 runs). Task rows report AP (↑), except fovea localization error (↓). $\Delta_{method}(\%)$: average relative gain over GD (↑). HV: hypervolume (↑).

Tasks	STL	GD	MMoE [KDD18]	TSσLN [TMLR26]	FAMO [NeurIPS23]	Nash-MTL [ICML22]	IMTL [ICLR21]	STCH [ICML24]	CAGrad [NeurIPS21]	FairGrad [ICML24]	selTG [ICLR25]	Ours
FA dataset												
DR	44.83±5.60	38.88±2.11	44.89±6.78	39.65±1.4	43.15±2.62	41.27±2.71	42.64±1.75	43.84±0.92	45.21±3.16	42.92±1.86	37.99±1.77	47.95±3.71
CNV	80.82±0.67	85.90±1.10	83.03±1.06	80.31±1.21	85.45±1.63	85.26±1.74	83.48±1.37	84.48±1.15	84.68±1.26	82.68±0.95	82.32±1.01	85.62±2.97
HyperF Type	51.96±0.77	52.37±1.86	52.82±2.8	49.58±1.38	53.43±1.09	54.21±2.34	51.10±0.98	52.88±1.24	51.70±2.59	46.76±1.43	51.62±1.11	52.53±2.01
HyperF Fovea	85.46±1.04	83.01±2.09	83.06±1.92	81.74±0.82	83.90±1.53	83.84±1.08	85.59±1.93	83.69±1.04	85.30±1.23	86.01±1.46	83.92±1.63	83.07±2.23
HyperF Area	78.64±0.84	78.75±1.51	76.43±1.96	77.87±1.45	79.30±0.90	79.39±1.64	77.94±0.95	78.83±1.27	78.37±1.03	75.05±2.79	78.85±2.01	78.73±1.57
HypoF Type	56.44±0.55	64.24±2.46	63.66±3.71	63.06±2.09	67.50±2.77	66.41±1.93	66.39±3.07	65.66±4.68	64.60±3.99	68.07±2.75	62.24±2.60	65.07±3.96
HypoF Fovea	44.97±2.62	44.82±2.14	45.88±1.62	44.94±0.6	45.35±2.80	47.18±2.55	43.20±1.94	45.06±2.72	44.86±2.13	43.35±1.61	44.09±0.94	49.91±2.94
HypoF Area	61.05±2.87	62.02±1.54	62.33±1.82	59.54±2.7	61.66±2.93	64.70±1.40	60.95±1.48	62.42±3.77	62.88±1.16	61.55±2.94	61.05±3.14	61.81±2.78
$\Delta_{method}(\%)$	-0.24	0	+1.51	-2.26	+2.49	+2.96	+0.57	+2.05	+2.24	-0.4	-1.62	**+4.46**
HV	-	0.028	0.035	0.036	0.037	0.036	0.027	0.034	0.034	0.027	0.024	**0.040**
AMD dataset												
AMD	84.81±3.77	86.88±2.62	86.79±2.48	68.42±6.08	82.31±5.67	83.13±2.25	49.09±5.04	81.96±4.73	87.97±2.82	81.69±5.51	72.53±12.12	84.32±4.61
fovea localization	2.88±0.26	3.56±0.30	3.3±0.34	3.24±0.05	3.41±0.11	3.48±0.23	3.28±0.06	3.50±0.19	2.98±0.38	2.89±0.33	3.45±0.17	3.35±0.05
drusen	72.82±7.66	57.71±16.57	73.42±3.68	40.73±12.31	58.18±14.10	67.14±16.60	27.83±2.88	57.33±7.06	80.55±5.78	34.21±5.36	40.49±9.95	76.54±6.63
exudate	63.93±6.66	63.91±8.58	64.74±11.56	57.11±4.2	61.59±6.92	50.46±7.57	39.77±12.15	64.26±6.32	63.19±7.29	65.66±6.49	48.61±7.42	80.35±6.28
hemorrhage	13.79±7.74	32.04±8.46	21.2±6.24	15.74±6.51	17.27±3.79	23.84±6.88	14.74±3.85	21.48±2.83	25.46±4.17	27.4±8.78	11.60±3.19	25.53±7.05
scar	32.84±11.43	47.37±21.28	61.45±9.32	35.33±17.34	36.48±11.99	59.09±18.45	14.58±7.26	52.39±11.78	65.09±4.92	31.2±9.1	38.53±15.63	65.15±9.27
$\Delta_{method}(\%)$	-7.46	0	+5.29	-21.43	-12.16	-1.24	-41.42	-4.39	+12.15	-12.28	-24.95	**+13.07**
HV	-	0.105	0.098	0.021	0.040	0.089	0.003	0.067	0.110	0.039	0.021	**0.162**

best baseline. The two metrics provide complementary views of the improvement. While Δ measures relative per-task gains, HV evaluates the joint quality of the obtained solutions. This distinction is particularly visible on AMD: CAGrad achieves a Δ closer to ours (+12.15% vs. +13.07%), but its HV remains notably lower (0.110 vs. 0.162), corresponding to a 47% higher HV for our method. In contrast, our method improves both metrics on both datasets, indicating that its gains translate into stronger joint multi-task performance.

Beyond the Medical Domain. We further evaluate on CelebA 9-task setting. Since several tasks already achieve high performance under GD, relative gains can disproportionately favor tasks with greater room for improvement. We therefore normalize gains by the remaining headroom of each task. In this setting, ASK-OT achieves a normalized average gain of $+5.05\%$, compared with $+2.25\%$ for IMTL, the next highest-performing baseline (Table 2). To assess scalability to a substantially larger number of tasks, we additionally evaluate ASK-OT in the full 40-task setting. This experiment compares ASK-OT with GD and examines whether the observed gains persist as the number of jointly learned tasks increases. ASK-OT achieves a normalized average gain of $+3.91\%$ over GD, improving 35 of 40 tasks (Figure 3).

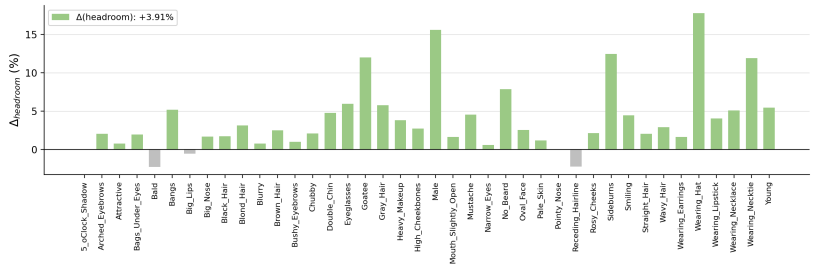


Figure 3: Task-wise normalized gains over GD on CelebA 40 tasks.

Table 2: $\Delta\%$ (headroom) on CelebA 9 tasks.

	Δ
MMoE	-0.72
TSσLN	-7.92
FAMO	0.89
Nash-MTL	0.01
IMTL	2.25
STCH	0.49
CAGrad	1.12
FairGrad	-14.39
selTG	-4.18
Ours	5.05

6.2 Analysis and Ablation Study

Knowledge modules specialization. Figure 5a tracks the mean pairwise cosine similarity between knowledge module outputs over training, comparing our method against uniform assignment. Under uniform setting, module similarities remain high and even increase over training, indicating that modules converge toward redundant, near-identical transformations which collapses the benefit of having multiple modules at all. With ASK-OT, similarities instead decrease steadily and stabilize at much lower values (≈ 0.15 for FA, ≈ 0.4 for AMD), showing that the learned assignment actively drives modules toward specialization rather than convergence.

Transport plan analysis. We assess where the learned optimal transport plan falls between two extremes: *full sharing*, in which every task draws equally on all modules, and

Table 3: Ablation study. Gains are computed relative to the GD baseline (first row).

KM	Routing	Δ on FA (%)	Δ on AMD (%)
✗	Uniform	0	0
✓	Uniform	-0.67%	+5.30%
✓	Softmax	+2.21%	+11.33%
✓	OT	-0.24%	+4.66%
✓	OT_{+MCC}	+2.09%	+7.78%
✓	$\text{OT}_{MCC+TCR}$	**+4.46%**	**+13.07%**
Linear	$\text{OT}_{MCC+TCR}$	+3.58%	-0.64%
MLP	$\text{OT}_{MCC+TCR}$	+2.10%	-5.49%

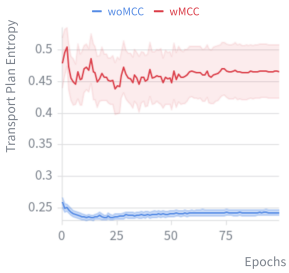
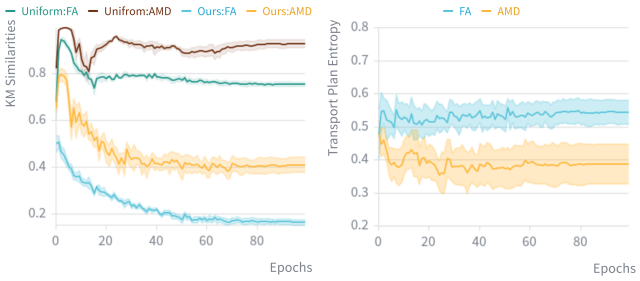


Figure 4: Transport plan entropy w/ and w/o MCC.

hard grouping, in which each task relies on an exclusive subset of modules. For task n, we compute the entropy of its column in the transport plan, $H(n) = -\sum_m \Pi_{mn} \log \Pi_{mn}$, and average across tasks to obtain $\bar{H} = \frac{1}{N}\sum_n H(n)$, which we normalize by the maximum attainable entropy: $\bar{H}_{\text{norm}} = (\bar{H}/\log M) \in [0, 1]$. A value near 1 indicates that tasks distribute their mass uniformly across modules, consistent with full sharing, while a value near 0 indicates that tasks concentrate their mass on a small subset of modules. Low entropy can result from either tasks specializing in different modules or all tasks using the same few modules. Since MCC ensures that every module contributes to multiple tasks (Lemma 1), the latter is ruled out. Thus, low entropy indicates task specialization across modules. Figure 5b reports $\bar{H}_{\text{norm}}$ over training. For both datasets, it settles early at intermediate values and remains stable, indicating that the model learns a soft assignment rather than collapsing to either full sharing or hard grouping (more analysis in Appendix F).



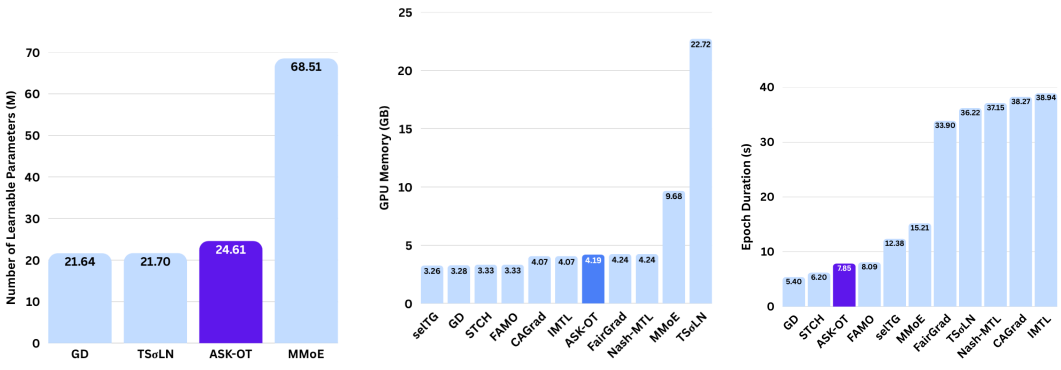
(a) (b)

Figure 5: (a) Mean pairwise cosine similarity between knowledge module outputs for ASK-OT and uniform assignment (lower indicate more specialized modules). (b) Normalized transport plan entropy $\bar{H}_{\text{norm}}$ of ASK-OT. 1 and 0 correspond to full sharing and hard grouping, respectively.

Ablation study. Table 3 progressively builds ASK-OT by adding each component to the GD baseline. Adding knowledge modules alone improves performance on AMD ($+5.30\%$) while slightly changing performance on FA (-0.67%), showing that added capacity alone is not sufficient. Routing with a per-task softmax over the same similarities (task-side marginal only) yields 2.21%, 11.33% on FA and AMD, below the strongest baselines, while balanced OT, which enforces both marginals as hard constraints, over-restricts the allocation (FA: -0.24%, AMD: $+4.66\%$). Adding *MCC* as shown in Figure 4 increases transport-plan entropy and promotes broader module utilization, further improving the gains (FA: $+2.09\%$, AMD: $+7.78\%$). Finally, adding *TCR* relaxes the task-side constraint and yields the best overall performance (FA: $+4.46\%$, AMD: $+13.07\%$).

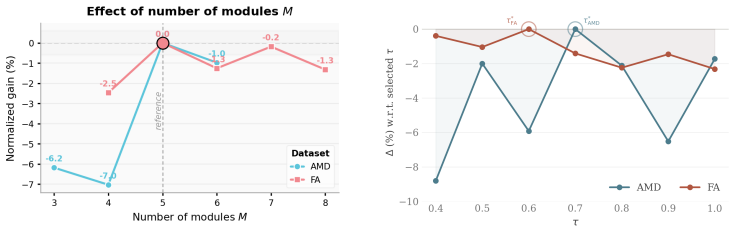
Architecture of the knowledge modules. Replacing attention-based $\mathcal{KM}$ with linear or MLP modules underperforms on both datasets (Table 3, bottom), with the larger gap for the MLP. This suggests that additional capacity alone does not necessarily improve knowledge sharing. One possible explanation lies in how freely each module can transform the shared representation. A linear module applies the same fixed projection to every input and cannot adapt to it. Self-attention and MLP modules are both non-linear and input-dependent, but self-attention builds its output as a weighted aggregation of the shared representation, which keeps it anchored to that representation. The MLP lacks this constraint and can alter the representation more freely, possibly moving away from the information useful for knowledge sharing.

Computational cost. Figure 6 reports epoch duration, GPU memory, and parameter count on the FA dataset. ASK-OT trains in 7.85s per epoch, close to GD (5.40s). Nash-MTL, our closest competitor



(a) (b) (c)

Figure 6: Computational cost comparison on the FA dataset. (a) Number of learnable parameters (M), reported only for GD and architecture-based methods, since the remaining baselines do not modify the underlying architecture. (b) GPU memory usage (GB). (c) Average epoch duration (s).



(a) (b)

Figure 7: Sensitivity to hyperparameters on FA and AMD. (a) Change in normalized gain as the number of knowledge modules M varies, relative to the default $M = 5$. (b) Normalized gain Δ (%) as a function of the TCR relaxation strength τ, relative to the selected τ for each dataset.

in performance, takes 37.15s epoch duration. Memory usage shows a similar trend: ASK-OT uses 4.19GB, comparable to the lightest baselines ($<$ 3.4GB) and well below MMoE (9.68GB) and TSσLN (22.72GB). Since ASK-OT adds modules, we also compare parameter count against GD and the other architecture-based methods. ASK-OT requires only $\sim$13% more than GD (21.64M) and TSσLN (21.70M), and substantially fewer than MMoE (68.51M). Overall, ASK-OT achieves its performance gains at a computational cost close to the simple GD baseline.

Effect of number of knowledge modules M. Figure 7a shows an asymmetric response around $M = 5$: reducing M causes sharp degradation (AMD -6.2% to -7.0%; FA -2.5%), while increasing M causes only mild loss. Intuitively, too few modules force tasks to compete for the same resources, whereas adding more modules leads to a loss that plateaus rather than growing with M.

TCR relaxation strength (τ). The parameter τ controls the strength of *TCR*. As $\tau \to 0$, the KL penalty vanishes and the allocation is determined primarily by the task–module cost C. As $\tau \to 1$, the task constraint approaches the exact target v_n. Figure 7b reports the normalized gain measured relative to the selected τ for each dataset. On both datasets, the best performance is obtained at an intermediate relaxation strength, with FA peaking at $\tau = 0.6$ and AMD at $\tau = 0.7$. The two datasets differ in sensitivity: FA is relatively robust, staying within 2.3% of its best value across the entire range, whereas AMD is considerably more sensitive, degrading by up to 8.8% at $\tau = 0.4$ and 6.5% at $\tau = 0.9$. Overall, the results favor a soft task-side constraint over either extreme and suggest selecting τ per dataset on the validation set.

7 Conclusion

We introduced ASK-OT, which formulates knowledge sharing in multi-task learning as a resource allocation problem solved with optimal transport. By combining learnable knowledge modules with a module-capacity constraint (MCC) and task constraint relaxation (TCR), ASK-OT enables continuous, asymmetric sharing without rigid task grouping, and improves over representative baselines on medical and general-domain benchmarks. Future work includes adding structure to the task representations, which are currently learned as free embeddings, to better capture task relationships.

AI USE STATEMENT

We used large language models (LLMs) for paper retrieval and discovery, as well as for editing support (clarity/grammar) and minor non-core code cleanup (formatting, docstrings, renaming, etc.). All LLM-assisted outputs and edits were reviewed by the authors, who remain fully responsible under the ICLR 2027 policy.

REFERENCES

Hao Ban and Kaiyi Ji. Fair resource allocation in multi-task learning. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=KLmWRMg6nL.

Felix JS Bragman, Ryutaro Tanno, Sebastien Ourselin, Daniel C Alexander, and M Jorge Cardoso. Learning task-specific and shared representations in medical imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 374–383. Springer, 2019.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. PLOT: Prompt learning with optimal transport for vision-language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of computation*, 87(314):2563–2609, 2018.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

Justin Engelmann and Miguel O Bernabeu. Training a high-performance retinal foundation model with half-the-data and 400 times less compute. *Nature Communications*, 16(1):6862, 2025.

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 109–117, 2004.

Huihui Fang, Fei Li, Huazhu Fu, Xu Sun, Xingxing Cao, Fengbin Lin, Jaemin Son, Sunho Kim, Gwenole Quellec, Sarah Matta, et al. Adam challenge: Detecting age-related macular degeneration from fundus images. *IEEE transactions on medical imaging*, 41(10):2828–2847, 2022.

Chuanwen Feng, Yilong Ren, and Xike Xie. Ot-filter: An optimal transport filter for learning with noisy labels. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16164–16174. IEEE, 2023.

Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34:27503–27516, 2021.

Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed Chi. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems*, 34:29335–29347, 2021.

Chi-Yao Huang, Khoa Vo, Aayush Verma, Duo Lu, and Yezhou Yang. Domain expansion: A latent space construction framework for multi-task learning. In *International Conference on Learning Representations*, volume 2026, pp. 101847–101857, 2026.

Wooseong Jeong and Kuk-Jin Yoon. Selective task group updates for multi-task optimization. In *International Conference on Learning Representations*, volume 2025, pp. 95046–95079, 2025.

Bin Li, Ye Shi, Qian Yu, and Jingya Wang. Unsupervised cross-domain image retrieval via prototypical optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3009–3017, 2024a.

Hongxia Li, Wei Huang, Jingya Wang, and Ye Shi. Global and local prompts cooperation via optimal transport for federated learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12151–12161. IEEE, 2024b.

Baijiong Lin and Yu Zhang. LibMTL: A Python library for multi-task learning. *Journal of Machine Learning Research*, 24(209):1–7, 2023.

Xi Lin, Xiaoyuan Zhang, Zhiyuan Yang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Smooth tchebycheff scalarization for multi-objective optimization. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=m4dO5L6eCp.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890, 2021a.

Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. Famo: Fast adaptive multitask optimization. *Advances in Neural Information Processing Systems*, 36:57226–57243, 2023.

Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International conference on learning representations*, 2021b.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1930–1939, 2018.

Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1851–1860. IEEE, 2019.

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*, pp. 16428–16446, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. Featured Certification.

Gabriel Peyré and Marco Cuturi. *Computational Optimal Transport*, volume 11 of *Foundations and Trends in Machine Learning*. Now Publishers, 2019.

Carl Sheridan, Adolfo Sebastiani, Claudio Campa, Francesco Parmeggiani, Carlo Incorvaia, Katia De Nadai, and Francesco Semeraro. Inflammatory mediators and angiogenic factors in choroidal neovascularization: Pathogenetic interactions and therapeutic implications. *Mediators of Inflammation*, 2010:1–14, 2010.

Mahshid Shiri, Cigdem Beyan, and Vittorio Murino. Madpot: Medical anomaly detection with clip adaptation and partial optimal transport. In *International Conference on Image Analysis and Processing*, pp. 247–259. Springer, 2025.

Mihai Suteu and Ovidiu Serban. Simplifying multi-task architectures through task-specific normalization. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

Piotr Szymański and Tomasz Kajdanowicz. A network perspective on stratification of multi-label data. In *First International Workshop on Learning with Imbalanced Domains: Theory and Applications*, pp. 22–35. PMLR, 2017.

Kim-Han Thung and Chong-Yaw Wee. A brief review on multi-task learning. *Multimedia Tools and Applications*, 77(22):29705–29725, 2018.

Cédric Villani and et al. *Optimal transport: old and new*, volume 338. 2008.

Pusheng Xu, Peranut Chotcomwongse, Weiyi Zhang, Xiaolan Chen, Xinyuan Wu, Florence HT Chung, Xueli Zhang, Mingguang He, Danli Shi, and Paisan Ruamviboonsuk. Angioreport: dataset and baseline methods for fundus angiography report generation. *British Journal of Ophthalmology*, 109(11):1283–1288, 2025.

Yan Zhao, Xiuying Wang, Tongtong Che, Guoqing Bao, and Shuyu Li. Multi-task deep learning for medical image computing and analysis: A review. *Computers in Biology and Medicine*, 153: 106496, 2023.

Gökhan Özbulak, Oscar Jimenez-del Toro, Maíra Fatoretto, Lilian Berton, and André Anjos. A multi-objective evaluation framework for analyzing utility-fairness trade-offs in machine learning systems. *Machine Learning for Biomedical Imaging*, 3(Special issue on FAIMI):938–957, 2025. doi: 10.59275/j.melba.2025-ab9a.

Appendix for

Learning How to Share: Optimal Transport for Medical Multi-Task Learning

Appendix organization:

A Task Relation Analysis **13**

B Method Details **14**

C Theoretical Insights: Asymmetric Sharing **15**

C.1 MCC: Enforcing Knowledge Sharing. 16

C.2 TCR: Modulating Knowledge Allocation. 17

C.3 Directional Asymmetry 18

D Theoretical Properties **19**

E Experimental Details **20**

F Transport plan Analysis. **21**

A Task Relation Analysis

This section provides the results supporting the task interaction analysis presented in Section 3.

Experimental setup. We investigate task interactions across three fluorescein angiography (FA)-based clinical tasks: CNV (binary classification), HyperF-Type (five-class classification), and HypoF-Type (three-class classification). We use RETFound-Green (Engelmann & Bernabeu, 2025) as the shared encoder, leveraging its domain-specific retinal pretraining to provide a meaningful representation for studying task relationships.

We compare single-task learning (STL) against all possible multi-task configurations among the three tasks. For each configuration, results are averaged over five random seeds. We report macro Area Under the Receiver Operating Characteristic curve (AUROC) and Average Precision (AP). In multi-task settings, all tasks share the encoder parameters while using independent linear classification heads as decoders. Models are optimized with AdamW using a batch size of 32, learning rate of 1.75×10^{-6}, and weight decay of 1.48×10^{-2}. We use a ReduceLROnPlateau scheduler with a factor of 0.8 and patience of 5 epochs.

Full results across metrics. Table 4 gives the full AUROC and AP results behind the transfer analysis in the main text, along with the expected (*exp.*) and observed (*Obser.*) direction of transfer for each configuration.

The *exp.* vs. *Obser.* columns show directly why clinical relatedness does not predict transfer well: most configurations expected to help (↑) only partly do, and several go the opposite way (↓) from what was expected. The clearest example is CNV–HypoF. HypoF is not expected to be a strong source task for CNV, yet CNV→HypoF gives the largest gain in the whole study (AUROC +2.52%, AP +3.69%). At the same time, the reverse direction, HypoF→CNV, actually hurts CNV (AUROC -0.76%, and the worst AP result in the table, -2.63%). This is the asymmetry described in the main text, shown numerically: the same pair is the best outcome for one task and one of the worst for the other.

Table 4: Macro AUROC, and AP for each task under every training configuration, mean ± std over 5 seeds. The *exp.* and *Obser.* columns indicate the expected transfer based on clinical relationships and the empirically observed change relative to STL, respectively.

Target task	configuration	AUC	AP	exp.	Obser.
CNV	STL	**91.80** ± 0.41	84.28 ± 0.62	-	-
	+ HyperF	91.57 ± 0.48	**85.68** ± 0.99	↑	↓↑
	+ HypoF	91.04 ± 0.52	81.65 ± 0.8	↓	↓↓
	+ HyperF & HypoF	90.54 ± 0.36	82.1 ± 1.04	mix	↓↓
HyperF	STL	82.87 ± 0.46	51.67 ± 0.8	-	-
	+ CNV	82.68 ± 0.41	52.17 ± 1.00	↑	↓↑
	+ HypoF	**83.35** ± 0.44	**53.4** ± 1.2	=/↑	↑↑
	+ CNV & HypoF	82.63 ± 0.58	52.1 ± 1.37	mix	↓↑
HypoF	STL	82.13 ± 0.57	55.4 ± 0.73	-	-
	+ CNV	**84.65** ± 1.05	**59.09** ± 1.8	↓	↑↑
	+ HyperF	81.81 ± 0.85	55.17 ± 0.54	=/↑	↓↓
	+ CNV & HyperF	81.54 ± 0.67	56.44 ± 0.4	mix	↓↓

AUROC and AP largely support the same overall pattern, with one minor exception. For CNV, HyperF is the only auxiliary task that improves AP relative to STL (+1.40%), while inducing a slight decrease in AUROC (-0.23%). This discrepancy motivates reporting both metrics: although they may differ in their assessment of small effects, they consistently support the larger patterns underlying our conclusions, including the CNV–HypoF asymmetry and the HyperF–HypoF interaction. Importantly, combining all auxiliary tasks consistently performs worse than the best individual task pairing on both metrics, further demonstrating that naive task grouping does not provide an effective strategy for knowledge sharing.

Notably, we additionally performed loss balancing to examine whether the observed interactions were driven by task loss weighting. The resulting transfer patterns remained consistent with the original objective, including the observed asymmetry between tasks.

B Method Details

Entropic OT via Sinkhorn. Directly solving the OT problem is time-consuming. Sinkhorn algorithm (Cuturi, 2013) uses an entropic constraint for optimization and uses the form $\Pi^\star = \mathrm{diag}(p)\, K\, \mathrm{diag}(q)$, where $K = \exp(-C/\epsilon)$ is the Gibbs kernel and $p \in \mathbb{R}_+^M$, $q \in \mathbb{R}_+^N$ are scaling vectors. Under the standard hard constraints $\Pi \mathbf{1}_N = \mathbf{u}$ and $\Pi^\top \mathbf{1}_M = \mathbf{v}$, p and q are found by alternately projecting onto each:

$$p \leftarrow \frac{\mathbf{u}}{Kq}, \qquad q \leftarrow \frac{\mathbf{v}}{K^\top p} \tag{5}$$

In practice we work in log-space for numerical stability (Peyré & Cuturi, 2019), tracking $u_{\log} = \log p$, $v_{\log} = \log q$ and replacing matrix-vector products with logsumexp operations.

Module Capacity Constraint and Task constraint Relaxation. Standard Sinkhorn updates both sides symmetrically. Our formulation keeps the module constraint (MCC) hard but for the task constraint (TCR) a soft KL penalty (equation 4) is used. Following the unbalanced OT (Chizat et al., 2018), a soft constraint with KL is enforced not by a full projection but by a *damped* projection. At each iteration the task side is updated first, followed by the module side using the newly updated task scaling:

$$v_{\log} \leftarrow \tau\Big(\log v - \mathrm{logsumexp}_m\big(M_r + u_{\log}\big)\Big), \qquad u_{\log} \leftarrow \log u - \mathrm{logsumexp}_n\big(M_r + v_{\log}\big) \tag{6}$$

where $M_r = -C/\epsilon$. Setting $\tau = 1$ recovers standard balanced Sinkhorn. We use this special case as a fast path with an off-the-shelf log-domain solver, and only run the custom damped iteration when $\tau < 1$.

Initialization. We initialize the module constraints as the uniform capacity vector $\mathbf{u} = \frac{N}{M}\mathbf{1}_M$ (MCC) and the task constraint as the uniform target $\mathbf{v} = \mathbf{1}_N$, so that total mass balances across both

Algorithm 1 The training process of ASK-OT

Input: Training data $\mathcal{X} = \{x\}$ with task labels, encoder f, number of knowledge modules M, number of tasks N, entropy parameter ϵ, task-relaxation strength $\tau \in [0, 1)$, maximum number of iterations for OT n_{iter}, number of epochs.
Output: The parameters of the f, knowledge modules $\mathcal{KM} = \{KM_m|_{m=1}^{M}\}$, task embeddings $\mathcal{T} = \{t_n|_{n=1}^{N}\}$, and task decoders Dec $= \{Dec_n|_{n=1}^{N}\}$.

Initialize $\mathcal{KM}, \mathcal{T}$, Dec.
Set module constraint $u = \frac{N}{M}\mathbf{1}_M$ and task constraint $v = \mathbf{1}_N$
for epochs **do**
  for each batch $x \in \mathcal{X}$ **do**
    Obtain the shared representation $h = f(x)$;
    Obtain module feature set $Z = \{z_m = KM_m(h)\}_{m=1}^{M}$;
    Calculate the cost matrix $C = 1 - Z^\top T$ and $M_r = -C/\epsilon$;
    Initialize $u_{\log}^{(0)} = \mathbf{0}_M$
    for $t_{\text{in}} = 1, 2, \ldots, n_{\text{iter}}$ **do**
      Update $v_{\log}^{(t_{\text{in}})} = \tau\Big(\log v - \text{logsumexp}_m\big(M_r + u_{\log}^{(t_{\text{in}}-1)}\big)\Big)$
      Update $u_{\log}^{(t_{\text{in}})} = \log u - \text{logsumexp}_n\big(M_r + v_{\log}^{(t_{\text{in}})}\big)$
    end for
    Obtain the transport plan as $\Pi^\star = \exp\big(M_r + u_{\log}^{(t_{\text{in}})}\mathbf{1}_N^\top + \mathbf{1}_M v_{\log}^{(t_{\text{in}})\top}\big)$
    Calculate task-specific representations $z'_n = \sum_{m=1}^{M} \Pi^\star_{mn} z_m$ for $n = 1, \ldots, N$
    Calculate task predictions $\hat{y}_n = Dec_n(z'_n)$ for $n = 1, \ldots, N$
    Update $f, \mathcal{KM}, \mathcal{T}, Dec$ with the task losses $\{\mathcal{L}_n(\hat{y}_n, y_n)|_{n=1}^{N}\}$;
  end for
end for
return $f, \mathcal{KM}, \mathcal{T}, Dec$

sides ($\sum_m u_m = \sum_n v_n = N$). The scaling variable is initialized at $u_{\log}^{(0)} = \mathbf{0}_M$ (i.e., $p = \mathbf{1}$ before the first update), which corresponds to $\Pi^{(0)} \propto K$.

Iterative updates. Given the cost matrix $C_{m,n} = 1 - \text{sim}(z_m, t_n)$ computed for a batch, the algorithm alternates the two updates above for a fixed number of iterations n_{iter} (we do not use a convergence-based stopping criterion, so as to keep the routing step of constant cost across batches). The final transport plan is recovered as

$$\Pi^\star = \exp\big(M_r + u_{\log}\mathbf{1}_N^\top + \mathbf{1}_M v_{\log}^\top\big) \tag{7}$$

The full pipeline during training and inference is presented in Algorithms 1 and 2

C Theoretical Insights: Asymmetric Sharing

This section provides the formal results underlying the asymmetric knowledge-sharing properties discussed in Section 5. We first show that *MCC* forces modules to share their capacity across tasks. We then characterize how *TCR* jointly modulates sharing according to module–task costs and task saturation, and finally show that the resulting directional sharing magnitudes need not be equal. Throughout, asymmetry refers to unequal sharing magnitudes and does not necessarily imply one-way transfer.

Assumption 1 (Cost coherence)**.** *Let $C_{:,n} = (c_{1n}, \ldots, c_{Mn})^\top \in \mathbb{R}^M$ denote the cost of task T_n against all modules. If two tasks T_i, T_j admit mutually beneficial knowledge transfer, while T_k is unrelated to both, then*

$$d\big(C_{:,i}, C_{:,j}\big) < d\big(C_{:,i}, C_{:,k}\big)$$

for a distance $d(\cdot, \cdot)$ induced by the same similarity used to construct C. That is, tasks that can benefit from one another are closer in cost space than either is to an unrelated task.

Assumption 2 (Faithful task representation)**.** *The learned embedding t_n correctly encodes what task T_n requires from the knowledge modules, such that $\text{sim}(z_m, t_n)$ is a faithful proxy for how well module m serves task T_n in the representation space.*

Algorithm 2 The inference process of ASK-OT

Input: Test input x, trained encoder f, trained knowledge modules $\mathcal{KM} = \{\mathcal{KM}_m|_{m=1}^{M}\}$, trained task embeddings $\mathcal{T} = \{t_n|_{n=1}^{N}\}$, trained task Decoders $\text{Dec} = \{Dec_n|_{n=1}^{N}\}$, entropy parameter ϵ

1: Set module constraint $u = \frac{N}{M}\mathbf{1}_M$ and task constraint $v = \mathbf{1}_N$
2: Obtain the shared representation $h = f(x)$;
3: Obtain module feature set $Z = \{z_m(h)|_{m=1}^{M}\}$;
4: Calculate the cost matrix $C = 1 - Z^\top T$ and $M_r = -C/\epsilon$;
5: Initialize $u_{\log}^{(0)} = \mathbf{0}_M$
6: **for** $t_{\text{in}} = 1, 2, \ldots, n_{\text{iter}}$ **do**
7: Update $v_{\log}^{(t_{\text{in}})} = \tau\Big(\log v - \text{logsumexp}_m\big(M_r + u_{\log}^{(t_{\text{in}}-1)}\big)\Big)$
8: Update $u_{\log}^{(t_{\text{in}})} = \log u - \text{logsumexp}_n\big(M_r + v_{\log}^{(t_{\text{in}})}\big)$
9: **end for**
10: Obtain the transport plan as $\Pi^\star = \exp\big(M_r + u_{\log}^{(t_{\text{in}})}\mathbf{1}_N^\top + \mathbf{1}_M v_{\log}^{(t_{\text{in}})\top}\big)$
11: Calculate task-specific representations $z'_n = \sum_{m=1}^{M} \Pi^\star_{mn}\, z_m$ for $n = 1, \ldots, N$
12: **return** $\{\hat{y}_n = Dec_n(z'_n)|_{n=1}^{N}\}$

Definition 1 (Preferred task)**.** *For a knowledge module m, its* preferred task *is the task with the lowest transport cost:*

$$PT(m) := \arg\min_n C_{m,n}$$

Definition 2 (Sharing amount)**.** *The amount of knowledge that task $T_{n'}$ contributes to task T_n is*

$$\Phi(n' \to n) := \sum_{m:\, PT(m)=n'} \Pi^*_{m,n}$$

Task T_n receives knowledge from *$T_{n'}$ whenever $\Phi(n' \to n) > 0$.*

C.1 MCC: Enforcing Knowledge Sharing.

Lemma (Forced spillover)**.** *Consider the hard-constrained limit $\tau \to 1$, in which the transport plan satisfies $\Pi^*\mathbf{1}_N = \mathbf{u}, \quad (\Pi^*)^\top\mathbf{1}_M = \mathbf{v}$, with $u_m = N/M$ and $v_n = 1$. Let $PT(m)$ denote the preferred task of module m, and define*

$$r_{PT(m)} := v_{PT(m)} - \sum_{j \neq m} \Pi^*_{j,PT(m)}$$

If $r_{PT(m)} < u_m$, then module m must allocate a positive amount of mass to at least one non-preferred task: $\sum_{n \neq PT(m)} \Pi^*_{m,n} > 0$.

Proof. By the row constraint imposed by MCC, and the hard column constraint:

$$\sum_{n=1}^{N} \Pi^*_{m,n} = u_m, \qquad \sum_{j=1}^{M} \Pi^*_{j,PT(m)} = v_{PT(m)}$$

Separating the contribution of module m from the remaining modules gives

$$\Pi^*_{m,PT(m)} = v_{PT(m)} - \sum_{j \neq m} \Pi^*_{j,PT(m)} = r_{PT(m)}$$

Therefore, if $r_{PT(m)} < u_m$, then $\Pi^*_{m,PT(m)} < u_m$. Since the total mass distributed by module m must equal u_m,

$$\sum_{n \neq PT(m)} \Pi^*_{m,n} = u_m - \Pi^*_{m,PT(m)} = u_m - r_{PT(m)} > 0$$

Hence, module m must allocate positive mass to one or more non-preferred tasks. □

Remark 1 (Spillover is unavoidable when $M < N$)**.** *Under the uniform constraints used in our formulation,*

$$u_m = \frac{N}{M}, \qquad v_n = 1$$

When $M < N$, each module must distribute more than one unit of mass: $u_m = \frac{N}{M} > 1$. On the other hand, for any module m, the remaining mass that its preferred task can receive satisfies

$$r_{PT(m)} = v_{PT(m)} - \sum_{j \neq m} \Pi^*_{j,PT(m)} \leq v_{PT(m)} = 1$$

Hence,

$$r_{PT(m)} \leq 1 < \frac{N}{M} = u_m$$

By Lemma 1, every module must therefore allocate positive mass to at least one non-preferred task. Thus, when $M < N$, knowledge sharing is not an exceptional outcome of the optimization: it is structurally enforced by MCC. Each module must distribute its capacity across multiple tasks, while the transport cost determines where this additional mass is allocated.

C.2 TCR: Modulating Knowledge Allocation.

Lemma (Relative sharing under soft TCR)**.** *Consider the soft TCR formulation with $\tau < 1$, and let $a_n = \sum_{m=1}^{M} \Pi^*_{mn}$ denote the total mass allocated to task T_n. For a fixed module m and two destination tasks T_i and T_j, the relative amount of mass allocated by module m to these tasks satisfies*

$$\frac{\Pi^*_{mi}}{\Pi^*_{mj}} = \exp\left(-\frac{C_{mi} - C_{mj}}{\epsilon}\right)\left(\frac{v_i/a_i}{v_j/a_j}\right)^{\frac{\lambda}{\epsilon}}, \qquad \lambda = \epsilon\frac{\tau}{1-\tau}$$

Consequently, $\Pi^*_{mi} > \Pi^*_{mj}$ if and only if

$$C_{mi} - C_{mj} < \lambda \log\left(\frac{v_i/a_i}{v_j/a_j}\right)$$

Thus, a lower module–task cost favors allocation to T_i, while the task-saturation term can strengthen or counteract this preference.

Proof. Under the soft TCR objective, the optimal transport plan has the form

$$\Pi^*_{mn} = \gamma_m \exp\left(-\frac{C_{mn}}{\epsilon}\right)\left(\frac{v_n}{a_n}\right)^{\lambda/\epsilon}$$

where $\gamma_m > 0$ is a row-normalization factor and $a_n = \sum_{m=1}^{M} \Pi^*_{mn}$. For a fixed module m, consider two candidate destination tasks T_i and T_j. Taking the ratio of their assigned masses gives

$$\frac{\Pi^*_{mi}}{\Pi^*_{mj}} = \frac{\gamma_m \exp(-C_{mi}/\epsilon)(v_i/a_i)^{\lambda/\epsilon}}{\gamma_m \exp(-C_{mj}/\epsilon)(v_j/a_j)^{\lambda/\epsilon}}$$

The row-normalization factor γ_m cancels, yielding

$$\frac{\Pi^*_{mi}}{\Pi^*_{mj}} = \exp\left(-\frac{C_{mi} - C_{mj}}{\epsilon}\right)\left(\frac{v_i/a_i}{v_j/a_j}\right)^{\lambda/\epsilon}$$

Taking logarithms:

$$\log\frac{\Pi^*_{mi}}{\Pi^*_{mj}} = -\frac{C_{mi} - C_{mj}}{\epsilon} + \frac{\lambda}{\epsilon}\log\left(\frac{v_i/a_i}{v_j/a_j}\right)$$

Therefore, $\Pi^*_{mi} > \Pi^*_{mj}$ if and only if

$$-(C_{mi} - C_{mj}) + \lambda \log\left(\frac{v_i/a_i}{v_j/a_j}\right) > 0$$

or equivalently: $C_{mi} - C_{mj} < \lambda \log\left(\frac{v_i/a_i}{v_j/a_j}\right)$. This expression separates the two factors governing the relative allocation. The first term, $C_{mi} - C_{mj}$ captures the module's preference between the two tasks: a smaller cost for T_i favors allocation to T_i. The second term, $\log\left(\frac{v_i/a_i}{v_j/a_j}\right)$ captures their relative saturation. If T_i is less supplied relative to its target, i.e.,

$$\frac{a_i}{v_i} < \frac{a_j}{v_j}$$

then $\frac{v_i/a_i}{v_j/a_j} > 1$, which further increases the relative allocation to T_i. Conversely, if T_i is more saturated than T_j, the saturation term counteracts its cost advantage.

Hence, under soft TCR, relative sharing is determined jointly by module–task affinity and global task saturation. The task-dependent saturation term introduced by TCR is important for asymmetric sharing because it allows the allocation preference of a module to depend not only on its task-specific cost, but also on how much mass the corresponding task has already received. □

C.3 Directional Asymmetry

Proposition (Decoupled directional sharing)**.** *For two distinct tasks T_i and T_j, the directional sharing magnitudes*

$$\Phi(i \to j) = \sum_{m:PT(m)=i} \Pi^*_{m,j}, \qquad \Phi(j \to i) = \sum_{m:PT(m)=j} \Pi^*_{m,i}$$

are governed by different preferred-module sets, module–task costs, and task-saturation factors. The optimization imposes no constraint requiring $\Phi(i \to j) = \Phi(j \to i)$. Hence, the two directional sharing magnitudes need not be equal.

Proof. From the KKT characterization of the optimal transport plan,

$$\Pi^*_{mn} = \gamma_m \exp\left(-\frac{C_{mn}}{\epsilon}\right)\left(\frac{v_n}{a_n}\right)^{\lambda/\epsilon}, \qquad a_n = \sum_{m=1}^{M} \Pi^*_{mn}$$

Consider first the flow from T_i to T_j. By definition,

$$\Phi(i \to j) = \sum_{m:PT(m)=i} \Pi^*_{mj}$$

Substituting the KKT form gives

$$\Phi(i \to j) = \left(\frac{v_j}{a_j}\right)^{\lambda/\epsilon} \sum_{m:PT(m)=i} \gamma_m \exp\left(-\frac{C_{mj}}{\epsilon}\right)$$

Thus, $\Phi(i \to j)$ is determined by the modules whose preferred task is T_i, the costs of assigning those modules to T_j, and the saturation factor of T_j. Similarly,

$$\Phi(j \to i) = \sum_{m:PT(m)=j} \Pi^*_{mi}$$

and therefore

$$\Phi(j \to i) = \left(\frac{v_i}{a_i}\right)^{\lambda/\epsilon} \sum_{m:PT(m)=j} \gamma_m \exp\left(-\frac{C_{mi}}{\epsilon}\right)$$

Hence, the reverse flow is governed by the modules whose preferred task is T_j, their costs for serving T_i, and the saturation factor of T_i.

These two expressions involve different preferred-module sets,

$$\mathcal{M}_i = \{m : PT(m) = i\}, \qquad \mathcal{M}_j = \{m : PT(m) = j\}$$

as well as different cross-task costs and saturation factors. Importantly, neither the row constraints $\Pi^* \mathbf{1}_N = \mathbf{u}$ nor the task-side TCR term imposes a relation of the form $\Phi(i \to j) = \Phi(j \to i)$. Consequently, equality of the two directional flows is not a structural property of the optimization problem. It can occur for particular configurations of the costs, module assignments, and saturation levels, but it is not enforced by the formulation. Therefore, the two directional sharing magnitudes need not be equal. □

D THEORETICAL PROPERTIES

The following propositions establish the well-posedness of the OT formulation and the module-utilization guarantee.

Well-posedness of the optimization. We first establish that, for any fixed cost matrix C and regularization strength $\epsilon > 0$, the proposed row-hard, column-soft transport formulation admits a unique optimal plan.

Proposition 2 (Existence and uniqueness)**.** *For any fixed cost matrix C and $\epsilon > 0$, the optimization problem*

$$\min_{\substack{\Pi \geq 0 \\ \Pi \mathbf{1}_N = u}} \langle \Pi, C \rangle - \epsilon H(\Pi) + \frac{1}{\tau'} \mathrm{KL}\big(\Pi^\top \mathbf{1}_M \,\big\|\, v\big) \tag{8}$$

admits a unique optimal transport plan Π^*, where $\tau' > 0$ denotes the weight of the task-side KL penalty.

Proof. The feasible set

$$\mathcal{F} = \big\{\Pi \in \mathbb{R}_+^{M \times N} : \Pi \mathbf{1}_N = u\big\} \tag{9}$$

is convex, being the intersection of the nonnegative orthant with an affine subspace. It is also closed, and bounded: since $\Pi_{mn} \geq 0$ and $\sum_{n=1}^{N} \Pi_{mn} = u_m$ for every row m, each entry satisfies $0 \leq \Pi_{mn} \leq u_m$. Closedness and boundedness in a finite-dimensional space imply that $\mathcal{F}$ is compact.

Next, we verify the strict convexity of the objective on $\mathcal{F}$. The transport cost $\langle \Pi, C \rangle$ is linear, hence convex. The entropic term

$$-\epsilon H(\Pi) = \epsilon \sum_{m,n} \Pi_{mn} \log \Pi_{mn} \tag{10}$$

is strictly convex on $\mathbb{R}_+^{M \times N}$, since $x \mapsto x \log x$ is strictly convex on $\mathbb{R}_+$ and the sum of strictly convex functions of independent coordinates is itself strictly convex. Finally, $\mathrm{KL}(\,\cdot\,\|v)$ is jointly convex in its first argument, and the map $\Pi \mapsto \Pi^\top \mathbf{1}_M$ is linear; the composition of a convex function with a linear map is convex, so the task-side penalty is convex in Π.

The objective is therefore a sum of a linear term, a strictly convex term, and a convex term, and is consequently strictly convex on $\mathcal{F}$. By compactness of $\mathcal{F}$ and continuity of the objective, a minimizer exists. By strict convexity of the objective on a convex domain, this minimizer is unique. Hence, for any fixed cost matrix C, the optimal transport plan Π^* is uniquely determined. □

Balanced knowledge utilization. We next show that the hard module-side constraint guarantees balance of module usage, independently of the task-side relaxation.

Proposition 3 (Guaranteed module usage)**.** *For any $\epsilon > 0$ and $\tau \in [0, 1)$, the transport plan produced after each inner iteration t_{in} of the Sinkhorn algorithm satisfies*

$$\sum_{n=1}^{N} \Pi_{mn}^{(t_{\mathrm{in}})} = u_m \qquad \textit{for every module } m. \tag{11}$$

In particular, under the uniform module constraint $u_m = N/M$, every knowledge module receives exactly the same total mass N/M at every iteration.

Proof. Within the Sinkhorn, the transport plan at iteration t_{in} is represented in log domain as

$$\Pi_{mn}^{(t_{\mathrm{in}})} = \exp\Big(M_{r,mn} + u_{\log,m}^{(t_{\mathrm{in}})} + v_{\log,n}^{(t_{\mathrm{in}})}\Big), \tag{12}$$

where M_r is the (fixed) log-domain cost kernel and $u_{\log}^{(t_{\mathrm{in}})}, v_{\log}^{(t_{\mathrm{in}})}$ are the current dual scaling vectors. Since the exponential function is strictly positive on $\mathbb{R}$, finiteness of $M_{r,mn}$, $u_{\log,m}^{(t_{\mathrm{in}})}$, and $v_{\log,n}^{(t_{\mathrm{in}})}$ immediately gives $\Pi_{mn}^{(t_{\mathrm{in}})} > 0$.

The module-side (row) update is applied last within each inner iteration and takes the closed-form projection

$$u_{\log,m}^{(t_{\text{in}})} = \log u_m - \text{logsumexp}_n \left(M_{r,mn} + v_{\log,n}^{(t_{\text{in}})} \right). \tag{13}$$

Substituting this into the row sum,

$$\begin{aligned} \sum_{n=1}^{N} \Pi_{mn}^{(t_{\text{in}})} &= \exp\Big(u_{\log,m}^{(t_{\text{in}})}\Big) \sum_{n=1}^{N} \exp\Big(M_{r,mn} + v_{\log,n}^{(t_{\text{in}})}\Big) \\ &= \frac{u_m}{\sum_{n=1}^{N} \exp\Big(M_{r,mn} + v_{\log,n}^{(t_{\text{in}})}\Big)} \cdot \sum_{n=1}^{N} \exp\Big(M_{r,mn} + v_{\log,n}^{(t_{\text{in}})}\Big) \qquad = u_m \end{aligned}$$

where the second line follows from $\exp(\log u_m - \text{logsumexp}_n(\cdot)) = u_m / \sum_n \exp(\cdot)$. The row constraint is thus satisfied exactly after every inner iteration, regardless of the current value of the task-side dual variables $v_{\log}^{(t_{\text{in}})}$ and independently of the damping parameter τ governing the task-side update. Setting $u_m = N/M$ for all m yields the stated uniform-usage guarantee. □

E Experimental Details

Datasets. We evaluate our method on multi-task benchmarks spanning medical and general computer vision domains. For medical evaluation, we use an 8-task fluorescein angiography (FA) dataset (Xu et al., 2025) and an AMD dataset (Fang et al., 2022). We further consider the CelebA benchmark (Liu et al., 2015) to evaluate our method beyond the medical domain. **(1) FA dataset.** We use the FA dataset of Xu et al. (2025). After filtering and label preprocessing, we retain 8 clinically motivated tasks: *CNV*, *HyperF/HypoF Fovea* (binary); *HypoF Type* and *HypoF/HyperF Area* (3-class); *HyperF Type* (5-class); and *Vascular abnormality (DR)* (4-class). All tasks have complete annotation. Data are split at the patient level (both eyes assigned jointly) into 70%/15%/15% train/validation/test, yielding 1234/267/270 examples (1771 total), using multi-label iterative stratification (Szymański & Kajdanowicz, 2017). **(2) AMD dataset.** We use the iChallenge-AMD/ADAM dataset (Fang et al., 2022) (400 colour fundus photographs; 89 AMD-positive, 311 AMD-negative). We derive 6 tasks: a global *AMD* diagnosis and four lesion-presence tasks (*Drusen*, *Exudate*, *Hemorrhage*, *Scar*), all binary, plus *Fovea localization* as 2D regression over pixel coordinates. All tasks have complete annotation; the binary labels are heavily imbalanced, with positive prevalence ranging from 22.2% (AMD) down to 3.2% (Scar). Data are split at the image level (70%/15%/15%) using multi-label iterative stratification (Szymański & Kajdanowicz, 2017) over the five binary labels. Fovea localization is excluded from the stratification matrix, since every image is annotated for this task. **(3) CelebA dataset.** We use the official release of CelebA, (202,599 face images, 10,177 identities), treating its 40 binary facial-attribute annotations as 40 independent classification tasks. We adopt CelebA's official train/validation/test partition (162,770/19,867/19,962 images).

Metrics. To assess task performance, we use metrics appropriate to each task type. For classification tasks Average Precision (AP) is reported due to class imbalance. For the regression task we report Mean Absolute Error (MAE). In addition, we compute HV using the `fairical` library (Özbulak et al., 2025), following its default conventions. All per-task scores are first normalized to $[0, 1]$, since HV is computed as a geometric volume over the joint metric space and an unnormalized axis would otherwise dominate the result for reasons unrelated to actual performance. Most task metrics are Average Precision (higher is better). The remaining regression metric (MAE, lower is better) is inverted as $1 - \text{MAE}$ prior to computing HV so that every axis follows the same higher-is-better convention required by the library. For each model, every seed's vector of per-task scores is treated as one point in the n-dimensional objective space.

Implementation Details. We use 5 seeds for the FA and AMD experiments, and 3 seeds for CelebA. All experiments are optimized using AdamW. The OT problem is solved with the entropic Sinkhorn algorithm (Cuturi, 2013) using 100 iterations. We use the logarithmic Sinkhorn implementation. Task decoders are linear heads throughout. For FA and AMD, RETFound-Green (Engelmann & Bernabeu, 2025) encoder and for CelebA, DINOv2-base (Oquab et al., 2024) is used. Full per-dataset hyperparameters are reported in Table 5. Note that τ and M were not tuned for the CelebA

Table 5: Training hyperparameters.

	FA / AMD	**CelebA**
Optimizer	AdamW	
Encoder lr	2×10^{-5}	
Module / decoder lr	2×10^{-4}	
Weight decay	1.883×10^{-3}	
LR scheduler	Warmup + cosine	
Sinkhorn iterations	100	
Task decoder	Linear	
Batch size	32	512
Backbone	RETFound-Green	DINOv2-base
Knowledge modules (M)	5	$5^{\dagger}$ (9 tasks)/$16^{\dagger}$ (40 tasks)
TCR relaxation strength (τ)	0.6 / 0.7	$0.7^{\dagger}$
Epochs	100	10
Image size	224^2	70^2

† Not tuned.

experiments. The models are trained using binary cross-entropy loss for multi-label classification tasks, cross-entropy loss for multi-class classification tasks, and MSE loss for the regression task.

We implemented the baseline methods using the LIBMTL library (Lin & Zhang, 2023) when available. For SelTG (Jeong & Yoon, 2025) and TSσLN (Suteu & Serban, 2026), we used the official implementations. All baselines were evaluated under the same experimental settings, including the shared encoder, task-specific decoders, loss functions, and training hyperparameters. For TSσLN, we evaluated the task-specific learning-rate multiplier over $5, 10, 20$ and selected the best-performing value, which was 10. For MMoE, we evaluated the number of experts over $3, 5, 8$ and selected the best-performing configuration. In both AMD and FA experiments, the best configuration used 3 experts.

Knowledge module architecture. Each knowledge module refines the encoder's token sequence via multi-head self-attention. Given an input sequence $\mathbf{x}$ consisting of a CLS token followed by patch tokens, the module applies standard multi-head self-attention (8 heads by default) over the full sequence, using $\mathbf{x}$ as query, key, and value simultaneously. Only the refined CLS token is retained from the attention output, which is then passed through a LayerNorm.

Task Embedding Initialization. Each t_n is initialized from a truncated normal distribution, $t_n \sim \mathcal{N}_{\text{trunc}}(0, 0.02^2)$, following standard initialization practice for learnable embeddings in vision transformers. We avoid near-zero initialization, which is common for classifier heads but unsuitable here: since t_n is used through cosine similarity, a near-zero norm causes its direction to be dominated by numerical noise, whereas excessively large initialization can allow Π to collapse to a near-deterministic assignment before the knowledge modules have specialized.

Data augmentation. For FA and AMD datasets, we apply mild geometric data augmentations while preserving clinically relevant structures. We randomly horizontally flip ($p = 0.5$), randomly rotate within $\pm 5^{\circ}$ with probability 0.25, and apply small affine transformations with translations up to 5% of the image size and scaling in the range $[0.95, 1.05]$ with probability 0.5. We do not apply any data augmentation for CelebA.

F TRANSPORT PLAN ANALYSIS.

The heatmaps in Figure 8 (FA) and Figure 9 (AMD) show the transport plan averaged over all validation samples, i.e., the empirical expectation $\mathbb{E}_x[\Pi(x)]$ of the input-dependent transport plan (across 5 seeds). Each heatmap is an $(M \times N)$ matrix, where entry (m, n) represents the mean transport mass assigned from knowledge module m to task n across the validation set. Since the module constraint is fixed and the task constraint is enforced according to the TCR constraint, the rows and columns reflect the corresponding aggregate module and task allocations, respectively. Thus, the averaged

transport plan provides an overview of the model's overall module-to-task sharing behavior across the validation samples.

Across five seeds on both datasets, the learned transport plans exhibit pronounced task-level asymmetry. Under MCC, each module has a fixed capacity of N/M, which can be allocated unevenly across tasks. A module may therefore concentrate its capacity on a task with greater learned demand or distribute it across multiple tasks, allowing the allocation to reflect task-specific representation requirements without imposing equal capacity across tasks. On FA dataset, DR draws the overwhelming majority of one module's entire capacity in every seed (entries $\approx$1.4–1.55) while receiving negligible mass (< 0.1) elsewhere. This is due to the fact that TCR allows a task's total allocation to exceed its target v_n. So, a high-demand task can draw most of a module's capacity, but MCC still forces the remainder to spill over to other tasks. Other tasks such as CNV instead spread their allocation across three to four modules with no entry exceeding roughly 0.5. On AMD dataset, drusen, hemorrhage, and scar classification each analogously drive one module to near-total capacity ($\approx$0.9–1.17) in every seed. This asymmetry extends to pairwise sharing: tasks that jointly rely on the same module never draw comparably from it, and the direction of imbalance is stable across seeds even as the identity of the shared module changes. On FA, HyperF-Type consistently draws several times more mass than CNV from a shared module (e.g. 0.84 vs. 0.22 in one seed), and HypoF Type and HypoF Area consistently outdraw CNV where they co-occupy a module (e.g. 0.50 and 0.67 vs. 0.15) — reproducing at the level of individual training runs the asymmetric CNV–HypoF interaction quantified in Table 4. On AMD dataset, exudate consistently receives a larger share than AMD classification from whichever module they jointly occupy (e.g. 0.80 vs. 0.28; 0.72 vs. 0.38; 0.65 vs. 0.38 across seeds). Together, these results show that ASK-OT does not split a shared module evenly between the tasks that use it. Each task's share is learned and it can be directional, non-symmetric transfer,which was the motivation of the method.

Beyond specialization, several tasks are softly grouped, drawing jointly on a shared module without either monopolizing it. On FA, the three hypofluorescence tasks (HypoF-Type, HypoF-Area, HypoF-Fovea) recurrently share a module in roughly comparable, moderate proportions (0.4–0.7 each) while still drawing smaller, non-negligible contributions (0.05–0.3) from the remaining modules, a soft rather than hard partition. On AMD dataset, a comparable pattern of low, uniform allocation appears for fovea localization, which receives minimal mass from every module in every seed (no entry exceeds 0.13, total allocation < 0.35) rather than concentrating on a specialized module. This is consistent with fovea localization being a lower-demand regression task relative to the five classification tasks in the benchmark, and shows that TCR can let a task draw a smaller total budget. These groupings emerge purely from learned interaction between modules and task representations, with no clinical taxonomy imposed on the routing mechanism: the AMD–exudate co-occupation happens to align with a known clinical relationship, while others, such as CNV–HyperF versus CNV–HypoF do not. This is consistent with our earlier finding that clinical relatedness is not a reliable predictor of transfer. ASK-OT groups tasks according to what the data supports rather than what taxonomy would suggest, sometimes recovering clinically sensible structure and sometimes not.

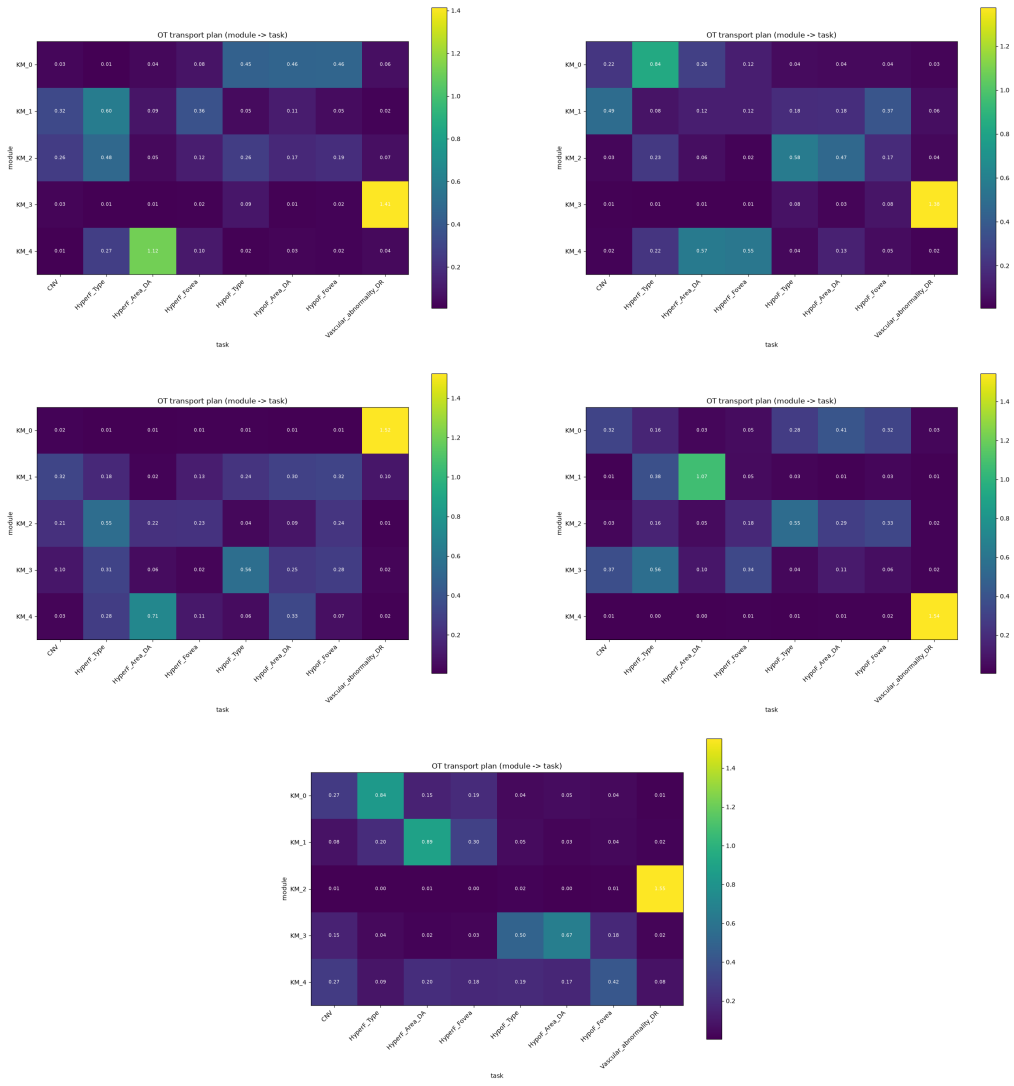


Figure 8: Each heatmap shows the transport plan averaged over all validation samples on the FA dataset, $\mathbb{E}_x[\Pi(x)]$, obtained from the best-validation checkpoint across different experiments.

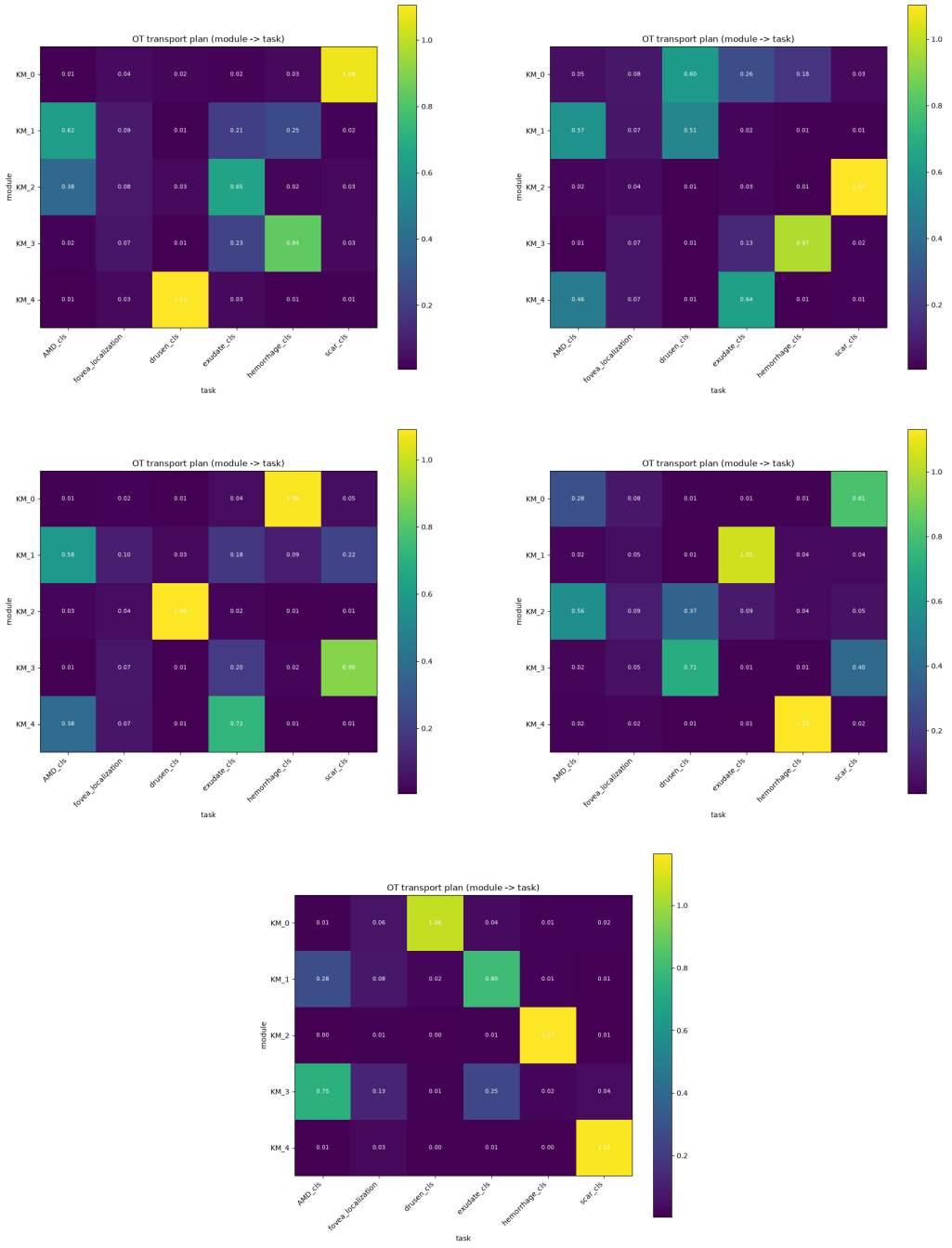


Figure 9: Each heatmap shows the transport plan averaged over all validation samples on the AMD dataset, $\mathbb{E}_x[\Pi(x)]$, obtained from the best-validation checkpoint across different experiments.